Introduction

2 minutes

Consider the scenario where you have imported data into Power BI from several different sources and, when you examine the data, it is not prepared for analysis. What could make the data unprepared for analysis?

When examining the data, you discover several issues, including:

- A column called **Employment status** only contains numerals.
- Several columns contain errors.
- Some columns contain null values.
- The customer ID in some columns appears as if it was duplicated repeatedly.
- A single address column has combined street address, city, state, and zip code.

You start working with the data, but every time you create visuals on reports, you get bad data, incorrect results, and simple reports about sales totals are wrong.

Dirty data can be overwhelming and, though you might feel frustrated, you decide to get to work and figure out how to make this data model as pristine as possible.

Fortunately, Power BI and Power Query offer you a powerful environment to clean and prepare the data. Clean data has the following advantages:

- Measures and columns produce more accurate results when they perform aggregations and calculations.
- Tables are organized, where users can find the data in an intuitive manner.
- Duplicates are removed, making data navigation simpler. It will also produce columns that can be used in slicers and filters.
- A complicated column can be split into two, simpler columns. Multiple columns can be combined into one column for readability.
- Codes and integers can be replaced with human readable values.

In this module, you will learn how to:

- Resolve inconsistencies, unexpected or null values, and data quality issues.
- Apply user-friendly value replacements.
- Profile data so you can learn more about a specific column before using it.
- Evaluate and transform column data types.
- Apply data shape transformations to table structures.
- Combine queries.
- Apply user-friendly naming conventions to columns and queries.
- Edit M code in the Advanced Editor.

Next unit: Shape the initial data

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

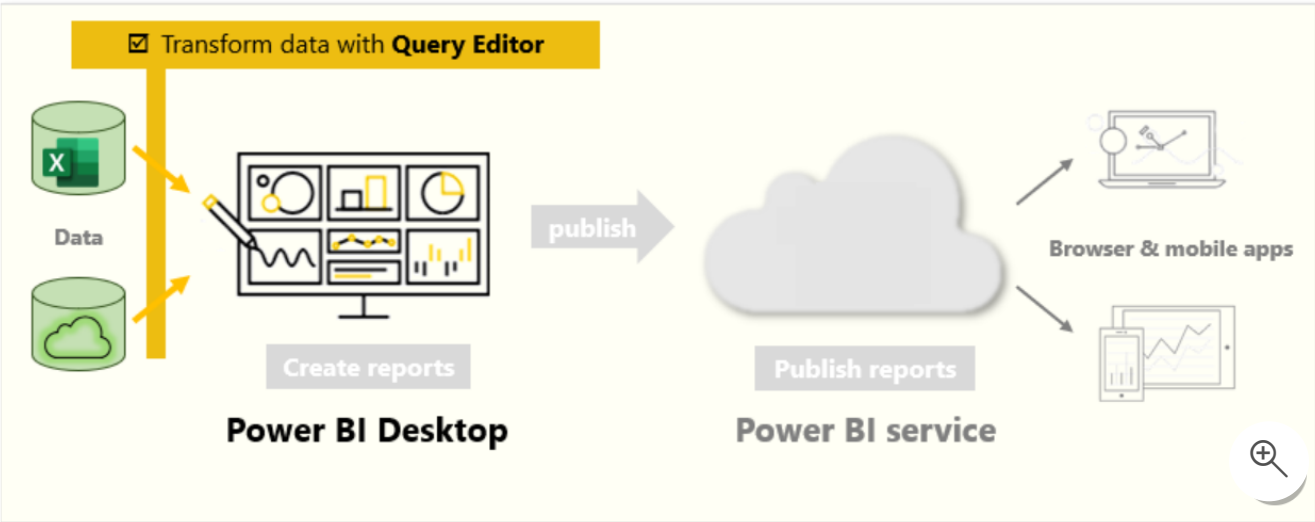
Shape the initial data

18 minutes

Power Query Editor in Power BI Desktop allows you to shape (transform) your imported data. You can accomplish actions such as renaming columns or tables, changing text to numbers, removing rows, setting the first row as headers, and much more. It is important to shape your data to ensure that it meets your needs and is suitable for use in reports.

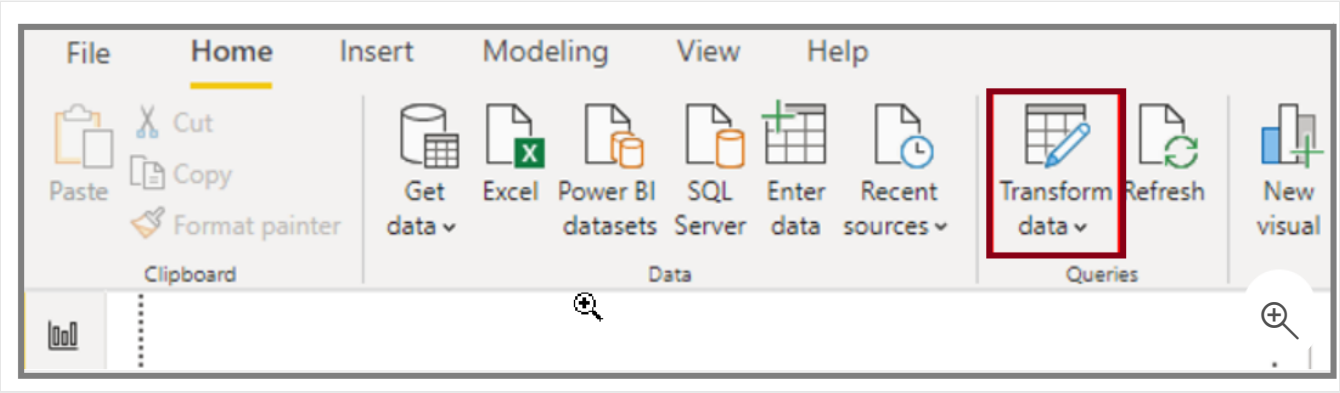
You have loaded raw sales data from two sources into a Power BI model. Some of the data came from a .csv file that was created manually in Microsoft Excel by the Sales team. The other data was loaded through a connection to your organization's Enterprise Resource Planning (ERP) system. Now, when you look at the data in Power BI Desktop, you notice that it's in disarray; some data that you don't need and some data that you do need are in the wrong format.

You need to use Power Query Editor to clean up and shape this data before you can start building reports.



Get started with Power Query Editor

To start shaping your data, open Power Query Editor by selecting the **Transform data** option on the **Home** tab of Power BI Desktop.



In Power Query Editor, the data in your selected query displays in the middle of the screen and, on the left side, the **Queries** pane lists the available queries (tables).

When you work in Power Query Editor, all steps that you take to shape your data are recorded. Then, each time the query connects to the data source, it automatically applies your steps, so your data is always shaped the way that you specified. Power Query Editor only makes changes to a particular view of your data, so you can feel confident about changes that are being made to your original data source. You can see a list of your steps on the right side of the screen, in the **Query Settings** pane, along with the query's properties.

The Power Query Editor ribbon contains many buttons you can use to select, view, and shape your data.

To learn more about the available features and functions, see The query ribbon.

> **Note**
>
> In Power Query Editor, the right-click context menus and **Transform** tab in the ribbon provide many of the same options.

Identify column headers and names

The first step in shaping your initial data is to identify the column headers and names within the data and then evaluate where they are located to ensure that they are in the right place.

In the following screenshot, the source data in the csv file for SalesTarget (sample not provided) had a target categorized by products and a subcategory split by months, both of which are organized into columns.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	ProductSubcategoryID	Name	January	February	March	April	May	June	July	August	September	October	November	December
1	1	Mountain Bikes	780000	790000	800000	810000	820000	830000	840000	850000	860000	870000	880000	890000
2	2	Road Bikes	4500	5000	5500	6000	6500	7000	7500	8000	8500	9000	9500	10000
3	3	Touring Bikes	501000	502000	503000	504000	505000	506000	507000	508000	509000	510000	511000	512000
4	4	Handlebars	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000	2100	2200
5	5	Bottom Brackets	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000	2100	2200
6	6	Brakes	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000	2100	00
7	7	Chains	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000	2100	0
8	8	Cranksets	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000	2100	00
9	9	Derailleurs	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000	2100	2200

However, you notice that the data did not import as expected.

	Column1	Column2	Column3	Column4	Column5	Column6	Column7
1			January	February	March	April	May
2							
3	ProductSubcategoryID	Name					
4	1	Mountain Bikes	780000	790000	800000	810000	820000
5	2	Road Bikes	4500	5000	5500	6000	6500
6	3	Touring Bikes	501000	502000	503000	504000	505000
7	4	Handlebars	1100	1200	1300	1400	1500
8	5	Bottom Brackets	1100	1200	1300	1400	1500
9	6	Brakes	1100	1200	1300	1400	1500
10	7	Chains	1100	1200	1300	1400	1500
11	8	Cranksets	1100	1200	1300	1400	1500
12	9	Derailleurs	1100	1200	1300	1400	1500

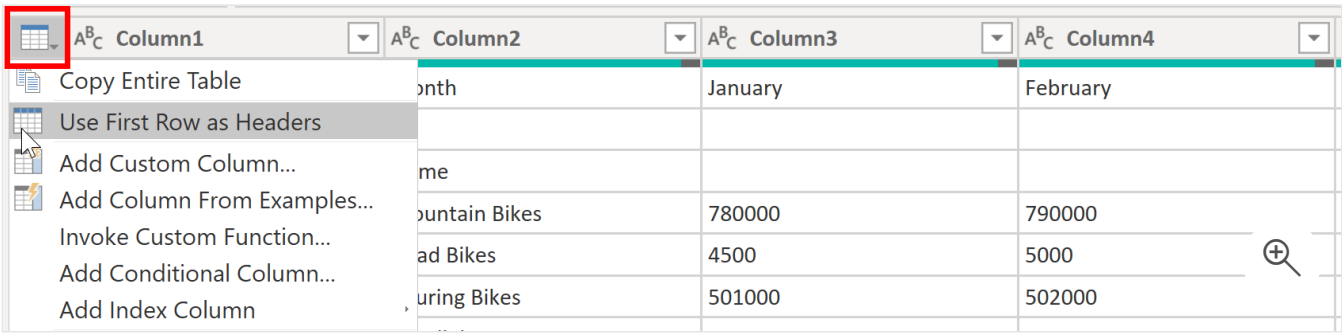
Consequently, the data is difficult to read. A problem has occurred with the data in its current state because column headers are in different rows (marked in red), and several columns have undescriptive names, such as **Column1**, **Column2**, and so on.

When you have identified where the column headers and names are located, you can make changes to reorganize the data.

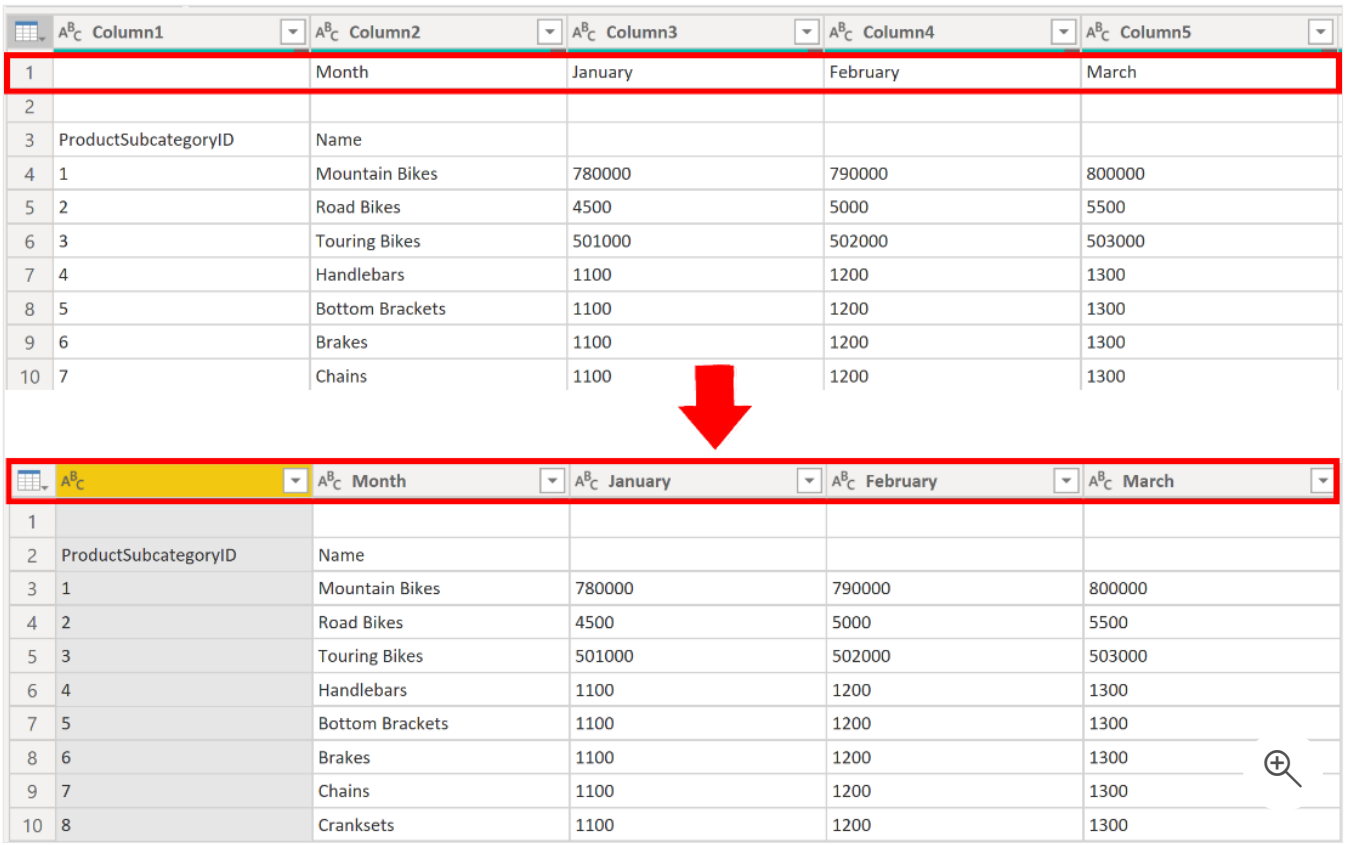
Promote headers

When a table is created in Power BI Desktop, Power Query Editor assumes that all data belongs in table rows. However, a data source might have a first row that contains column names, which is what happened in the previous SalesTarget example. To correct this inaccuracy, you need to promote the first table row into column headers.

You can promote headers in two ways: by selecting the **Use First Row as Headers** option on the **Home** tab or by selecting the drop-down button next to **Column1** and then selecting **Use First Row as Headers**.



The following image illustrates how the **Use First Row as Headers** feature impacts the data:



	Column1	Column2	Column3	Column4	Column5
1		Month	January	February	March
2					
3	ProductSubcategoryID	Name			
4	1	Mountain Bikes	780000	790000	800000
5	2	Road Bikes	4500	5000	5500
6	3	Touring Bikes	501000	502000	503000
7	4	Handlebars	1100	1200	1300
8	5	Bottom Brackets	1100	1200	1300
9	6	Brakes	1100	1200	1300
10	7	Chains	1100	1200	1300

		Month	January	February	March
1					
2	ProductSubcategoryID	Name			
3	1	Mountain Bikes	780000	790000	800000
4	2	Road Bikes	4500	5000	5500
5	3	Touring Bikes	501000	502000	503000
6	4	Handlebars	1100	1200	1300
7	5	Bottom Brackets	1100	1200	1300
8	6	Brakes	1100	1200	1300
9	7	Chains	1100	1200	1300
10	8	Cranksets	1100	1200	1300

Rename columns

The next step in shaping your data is to examine the column headers. You might discover that one or more columns have the wrong headers, a header has a spelling error, or the header naming convention is not consistent or user-friendly.

Refer to the previous screenshot, which shows the impact of the **Use First Row as Headers** feature. Notice that the column that contains the subcategory **Name** data now has **Month** as its column header. This column header is incorrect, so it needs to be renamed.

You can rename column headers in two ways. One approach is to right-click the header, select **Rename**, edit the name, and then press **Enter**. Alternatively, you can double-click the column header and overwrite the name with the correct name.

You can also work around this issue by removing (skipping) the first two rows and then renaming the columns to the correct name.

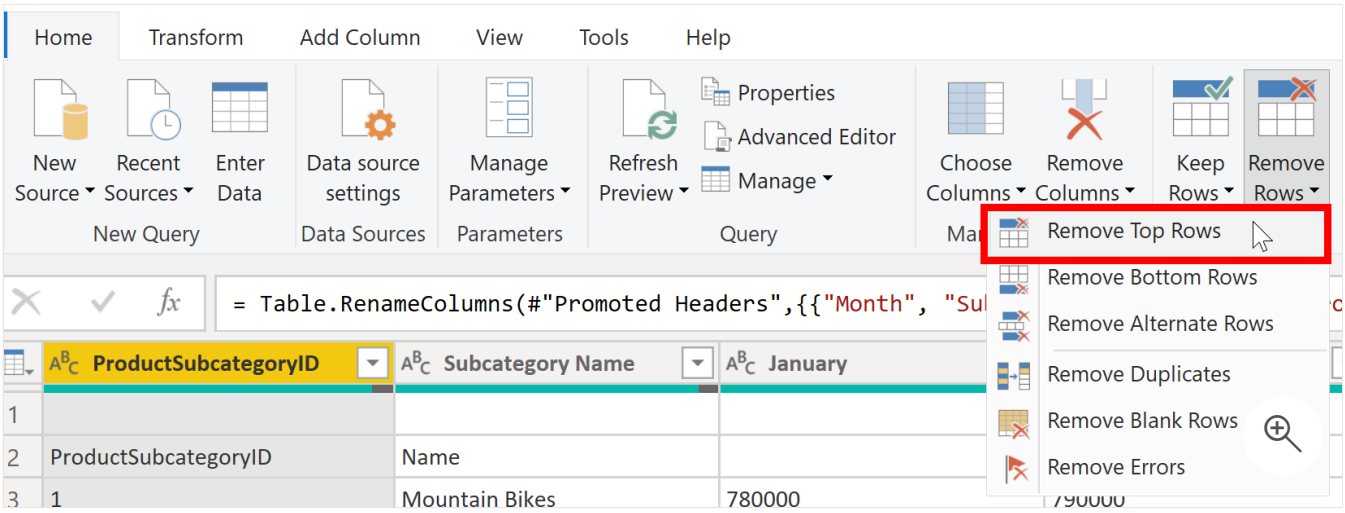
Remove top rows

When shaping your data, you might need to remove some of the top rows, for example, if they are blank or if they contain data that you do not need in your reports.

Continuing with the SalesTarget example, notice that the first row is blank (it has no data) and the second row has data that is no longer required.

	ProductSubcategoryID	Subcategory Name	January	February	March
1					
2	ProductSubcategoryID	Name			
3	1	Mountain Bikes	780000	790000	800000
4	2	Road Bikes	4500	5000	5500
5	3	Touring Bikes	501000	502000	503000
6	4	Handlebars	1100	1200	1300

To remove these excess rows, select **Remove Rows** > **Remove Top Rows** on the **Home** tab.



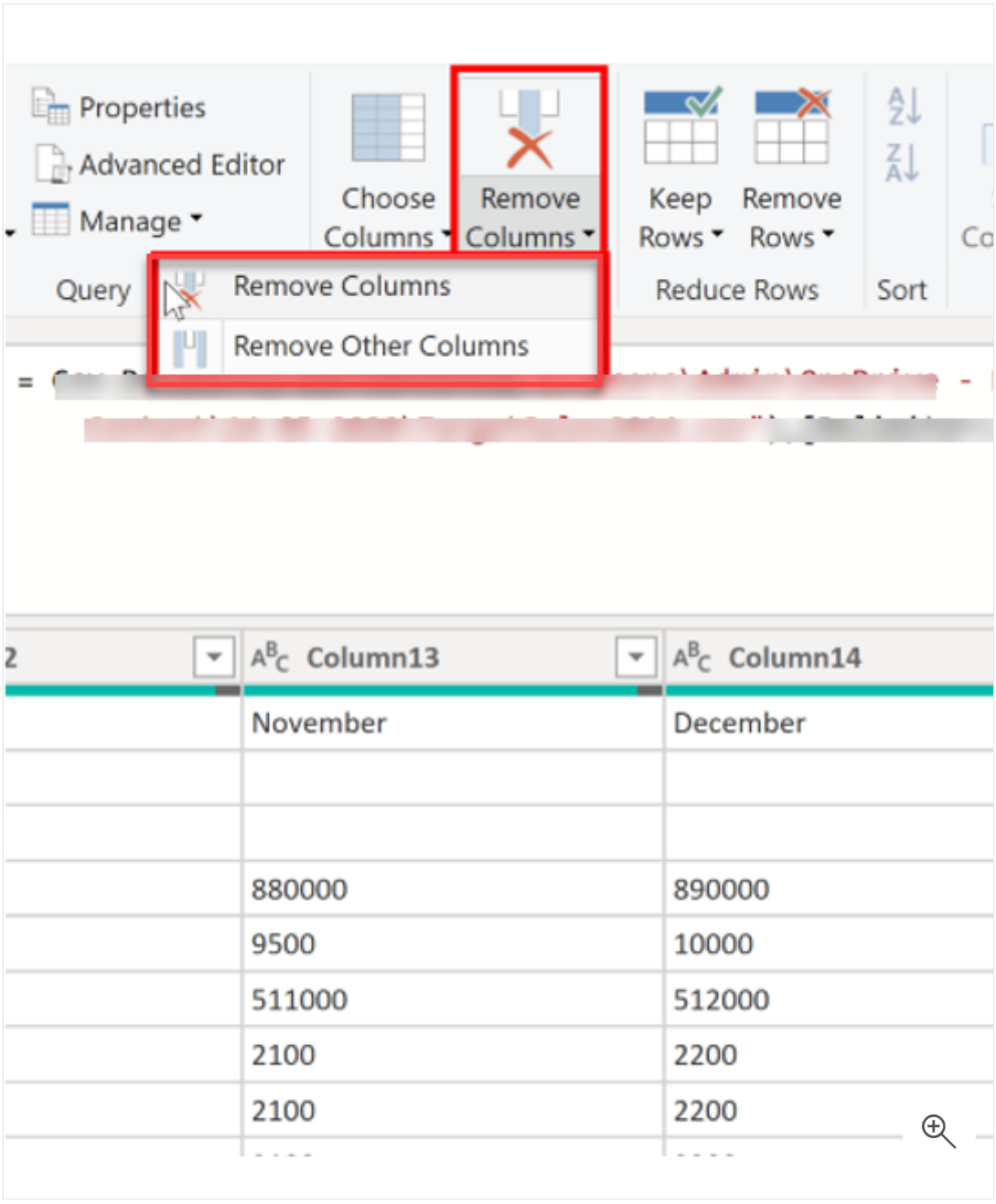
Remove columns

A key step in the data shaping process is to remove unnecessary columns. It is much better to remove columns as early as possible. One way to remove columns would be to limit the column when you get data from data source. For instance, if you are extracting data from a relational database by using SQL, you would want to limit the column that you extract by using a column list in the SELECT statement.

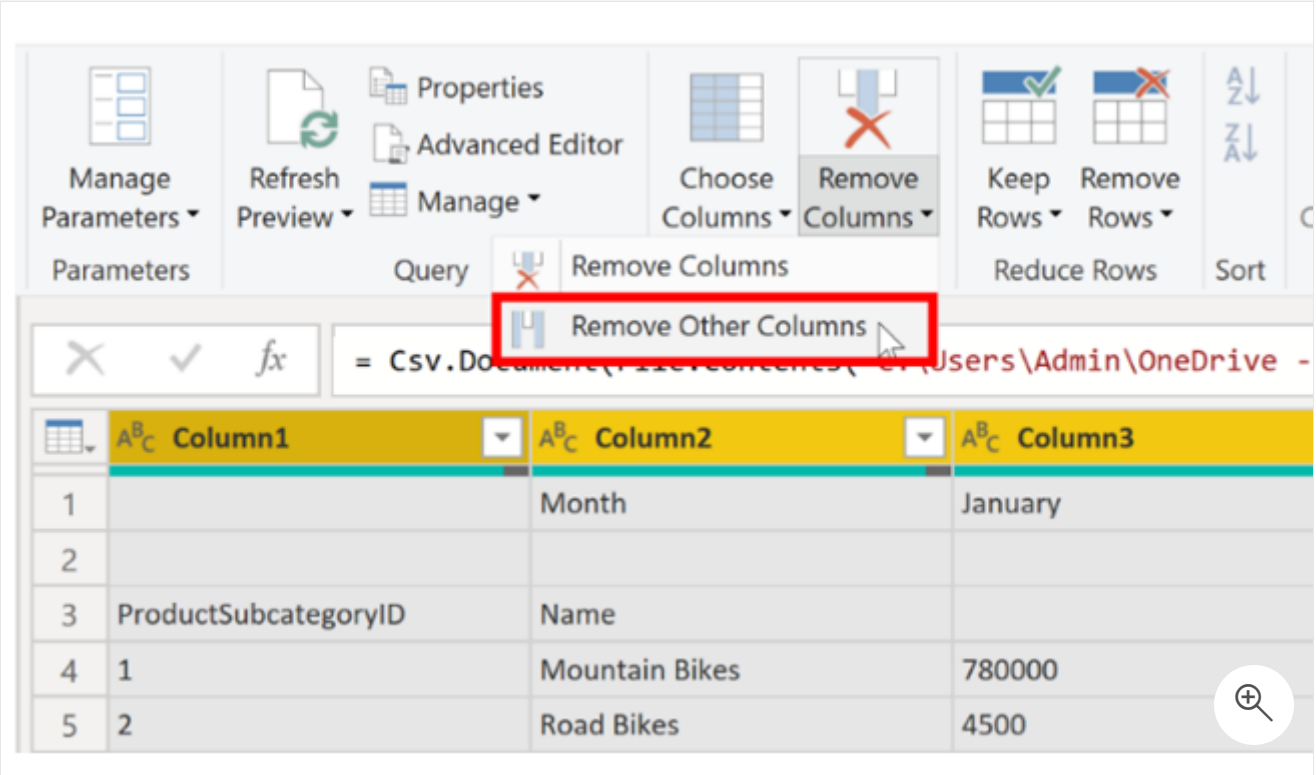
Removing columns at an early stage in the process rather than later is best, especially when you have established relationships between your tables. Removing unnecessary columns will help you to focus on the data that you need and help improve the overall performance of your Power BI Desktop datasets and reports.

Examine each column and ask yourself if you really need the data that it contains. If you don't plan on using that data in a report, the column adds no value to your data model. Therefore, the column should be removed. You can always add the column later, if your requirements change over time.

You can remove columns in two ways. The first method is to select the columns that you want to remove and then, on the **Home** tab, select **Remove Columns**.



Alternatively, you can select the columns that you want to keep and then, on the **Home** tab, select **Remove Columns** > **Remove Other Columns**.



Unpivot columns

Unpivoting is a useful feature of Power BI. You can use this feature with data from any data source, but you would most often use it when importing data from Excel. The following example shows a sample Excel document with sales data.

Year	2018	2019
January	$ 15,370	$ 16,063
February	$ 15,950	$ 12,161
March	$ 13,862	$ 14,180
April	$ 18,530	$ 6,516
May	$ 5,203	$ 19,395
June	$ 5,928	$ 19,324
July	$ 14,736	$ 15,939
August	$ 6,243	$ 15,390
Septemb	$ 15,178	$ 17,832
October	$ 18,148	$ 5,185
Novembe	$ 8,014	$ 9,[illegible]
Decembe	$ 19,470	$ 14[illegible]

Though the data might initially make sense, it would be difficult to create a total of all sales combined from 2018 and 2019. Your goal would then be to use this data in Power BI with three columns: **Month**, **Year**, and **SalesAmount**.

When you import the data into Power Query, it will look like the following image.

	Year	2018	2019
1	January	15370	16063
2	February	15950	12161
3	March	13862	14180
4	April	18530	6516
5	May	5203	19395
6	June	5928	19324
7	July	14736	15939
8	August	6243	15390
9	September	15178	17832
10	October	18148	5185
11	November	8014	9299
12	December	19470	1

Next, rename the first column to **Month**. This column was mislabeled because that header in Excel was labeling the 2018 and 2019 columns. Highlight the 2018 and 2019 columns, select the **Transform** tab in Power Query, and then select **Unpivot**.

	Year	Attribute	Value
	January	2018	15370
	January	2019	16063
	February	2018	15950
	February	2019	12161
	March	2018	13862
	March	2019	14180
	April	2018	18530
	April	2019	6516
	May	2018	5203
0	May	2019	19395
1	June	2018	5928
2	June	2019	19324
3	July	2018	14736
4	July	2019	15939
5	August	2018	6243
5	August	2019	15390
7	September	2018	15178
8	September	2019	17832
9	October	2018	18148
0	October	2019	5185
1	November	2018	8014
2	November	2019	9299
3	December	2018	19470
4	December	2019	1408

You can rename the **Attribute** column to **Year** and the **Value** column to **SalesAmount**.

Unpivoting streamlines the process of creating DAX measures on the data later. By completing this process, you have now created a simpler way of slicing the data with the **Year** and **Month** columns.

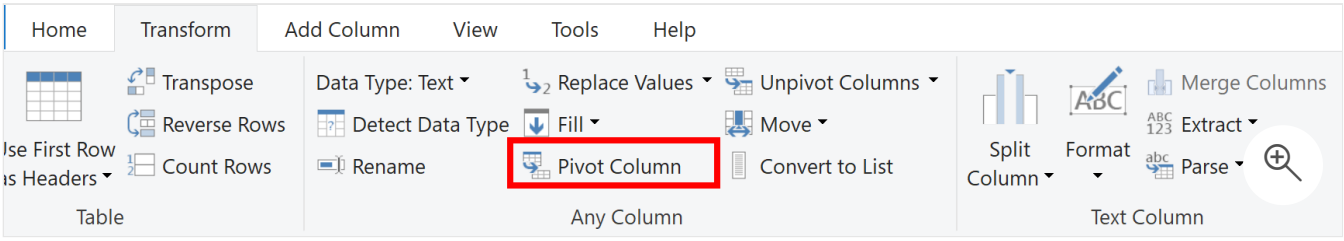
Pivot columns

If the data that you are shaping is flat (in other words, it has lot of detail but is not organized or grouped in any way), the lack of structure can complicate your ability to identify patterns in the data.

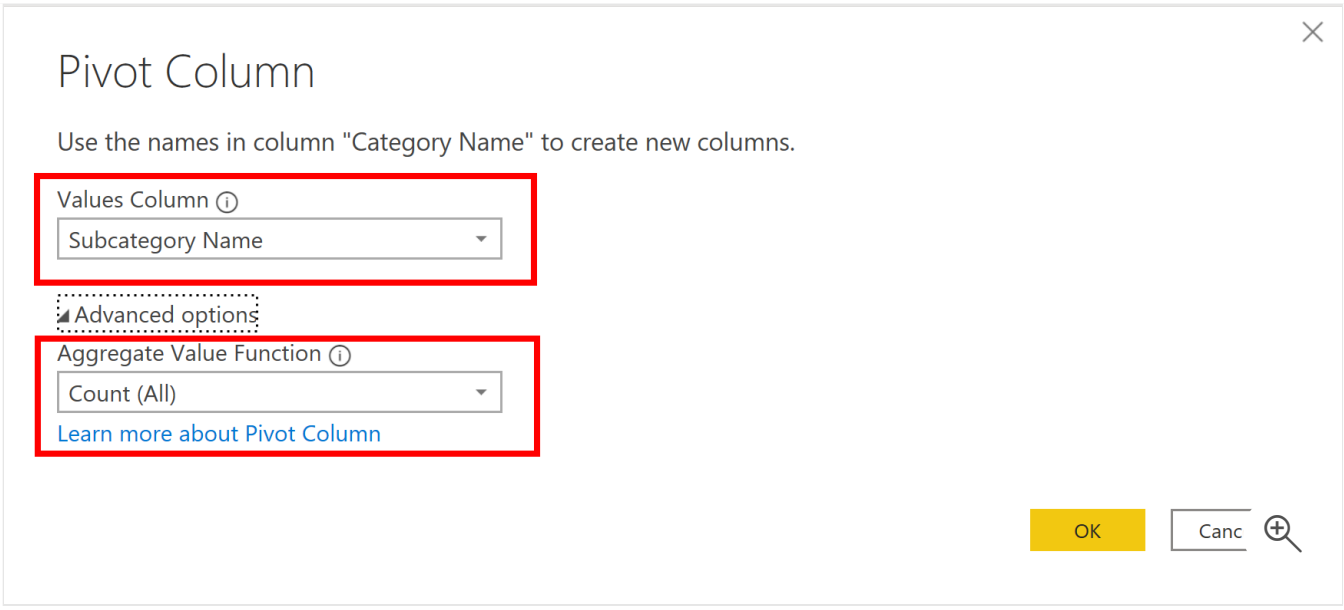
You can use the **Pivot Column** feature to convert your flat data into a table that contains an aggregate value for each unique value in a column. For example, you might want to use this feature to summarize data by using different math functions such as **Count**, **Minimum**, **Maximum**, **Median**, **Average**, or **Sum**.

In the SalesTarget example, you can pivot the columns to get the quantity of product subcategories in each product category.

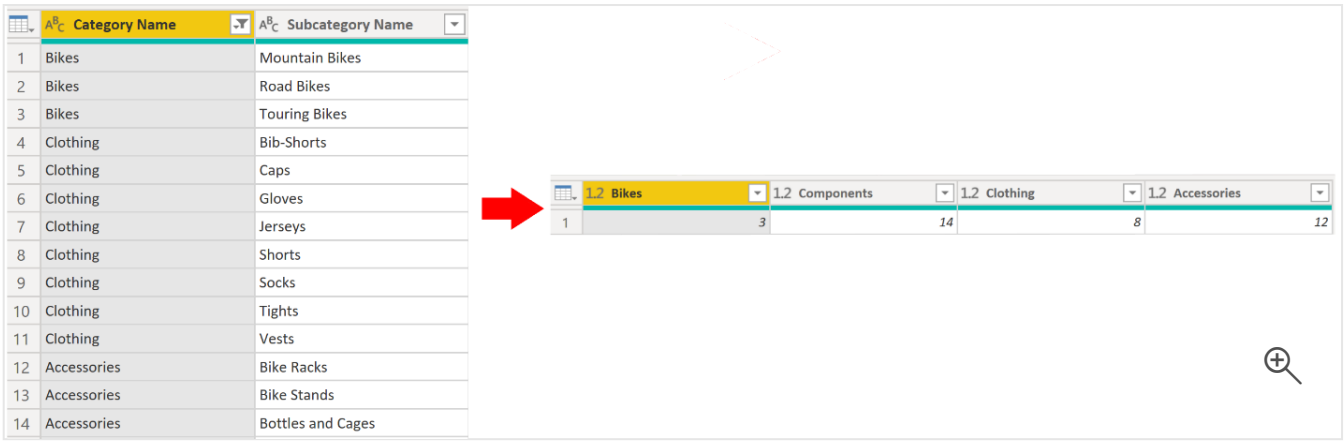
On the **Transform** tab, select **Transform** > **Pivot Columns**.



On the **Pivot Column** window that displays, select a column from the **Values Column** list, such as **Subcategory name**. Expand the advanced options and select an option from the **Aggregate Value Function** list, such as **Count (All)**, and then select **OK**.



The following image illustrates how the **Pivot Column** feature changes the way that the data is organized.



	Category Name	Subcategory Name
1	Bikes	Mountain Bikes
2	Bikes	Road Bikes
3	Bikes	Touring Bikes
4	Clothing	Bib-Shorts
5	Clothing	Caps
6	Clothing	Gloves
7	Clothing	Jerseys
8	Clothing	Shorts
9	Clothing	Socks
10	Clothing	Tights
11	Clothing	Vests
12	Accessories	Bike Racks
13	Accessories	Bike Stands
14	Accessories	Bottles and Cages

	Bikes	Components	Clothing	Accessories
1	3	14	8	12

Power Query Editor records all steps that you take to shape your data, and the list of steps are shown in the **Query Settings** pane. If you have made all the required changes, select **Close & Apply** to close Power Query Editor and apply your changes to your data model. However, before you select **Close & Apply**, you can take further steps to clean up and transform your data in Power Query Editor. These additional steps are covered later in this module.

Next unit: Simplify the data structure

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Simplify the data structure

8 minutes

When you import data from multiple sources into Power BI Desktop, the data retains its predefined table and column names. You might want to change some of these names so that they are in a consistent format, easier to work with, and more meaningful to a user. You can use Power Query Editor in Power BI Desktop to make these name changes and simplify your data structure.

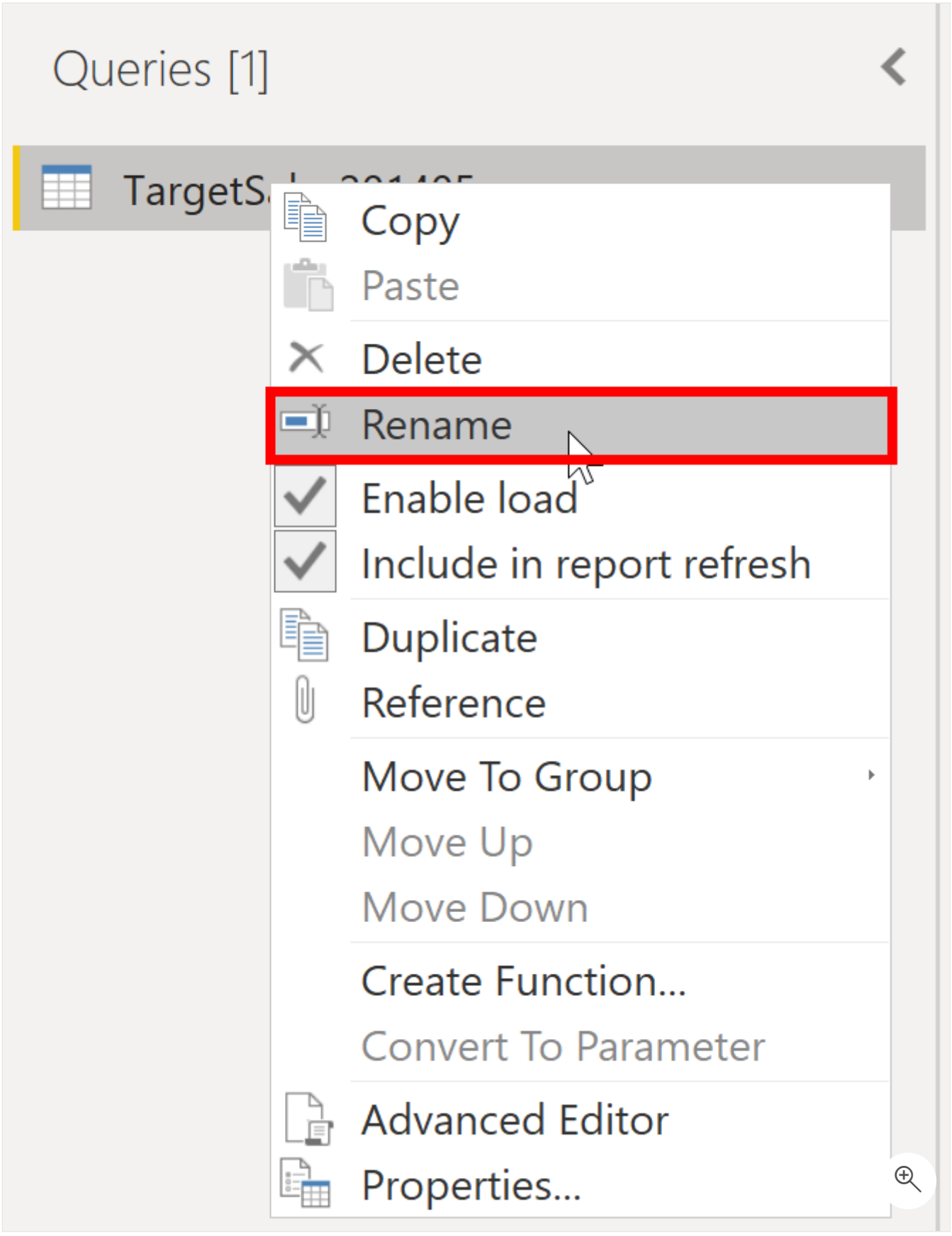
To continue with the previous scenario where you shaped the initial data in your model, you need to take further action to simplify the structure of the sales data and get it ready for developing reports for the Sales team. You have already renamed the columns, but now you need to examine the names of the queries (tables) to determine if any improvements can be made. You also need to review the contents of the columns and replace any values that require correction.

Rename a query

It's good practice to change uncommon or unhelpful query names to names that are more obvious or that the user is more familiar with. For instance, if you import a product fact table into Power BI Desktop and the query name displays as *FactProductTable*, you might want to change it to a more user-friendly name, such as *Products*. Similarly, if you import a view, the view might have a name that contains a prefix of *v*, such as *vProduct*. People might find this name unclear and confusing, so you might want to remove the prefix.

In this example, you have examined the name of the TargetSales query and realize that this name is unhelpful because you'll have a query with this name for every year. To avoid confusion, you want to add the year to the query name.

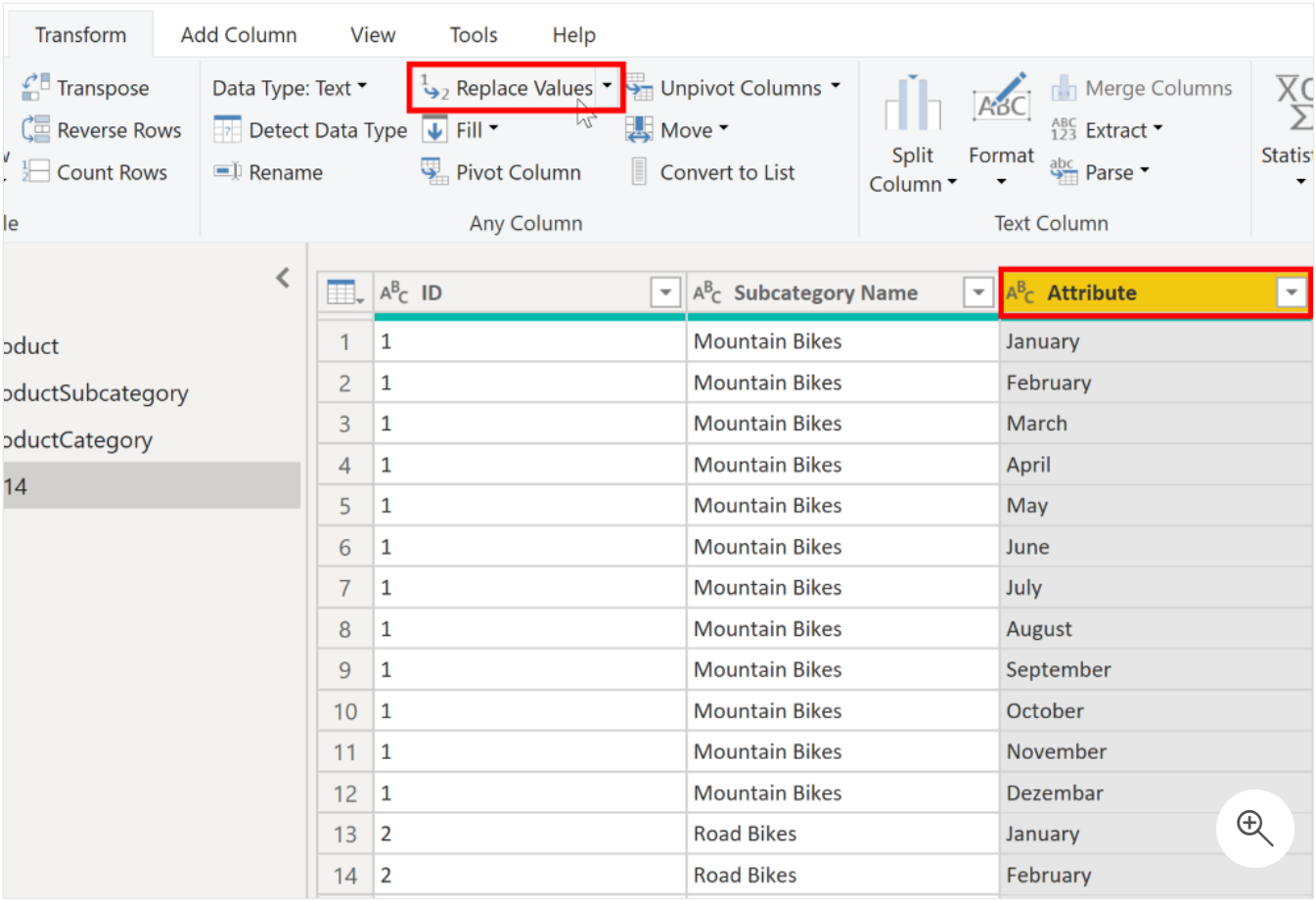
In Power Query Editor, in the **Queries** pane to the left of your data, select the query that you want to rename. Right-click the query and select **Rename**. Edit the current name or type a new name, and then press **Enter**.



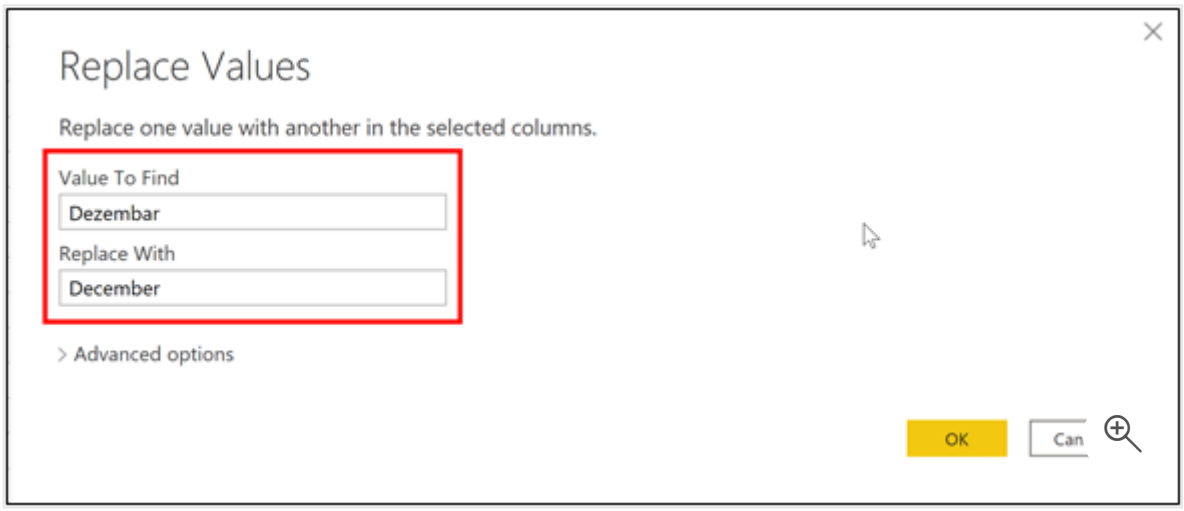
Replace values

You can use the **Replace Values** feature in Power Query Editor to replace any value with another value in a selected column.

In this example, you notice that, in the **Attribute** column, the month December is misspelled. You need to correct this spelling mistake. Select the column that contains the value that you want to replace (**Attribute** in this case), and then select **Replace Values** on the **Transform** tab.



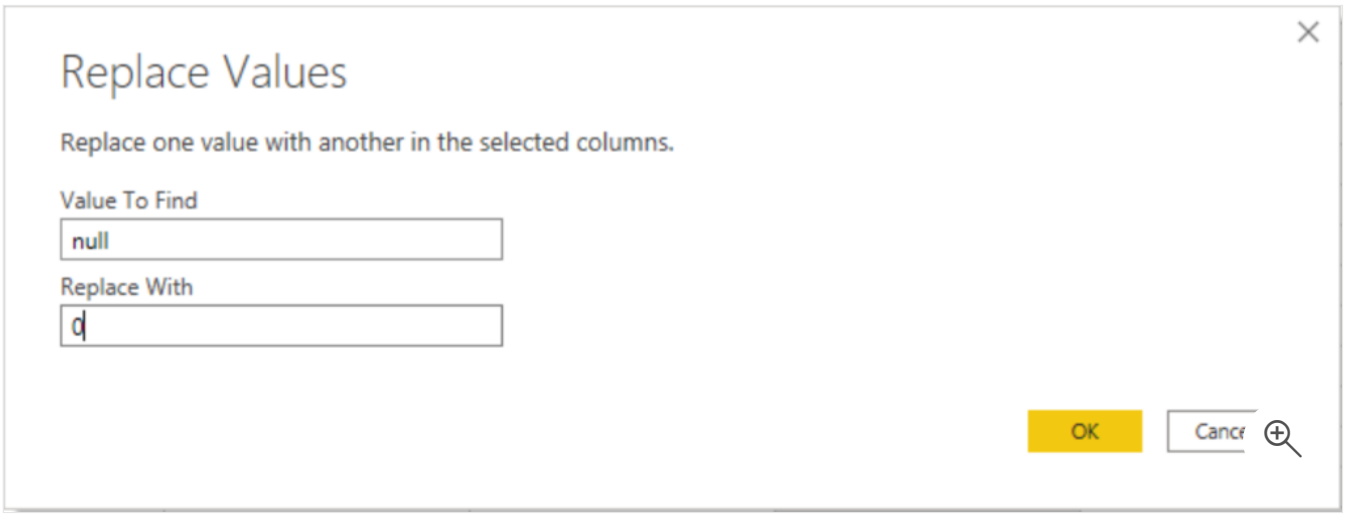
In the **Value to Find** box, enter the name of the value that you want to replace, and then in the **Replace With** box, enter the correct value name and then select **OK**. In Power Query, you can't select one cell and change one value, like you might have done in Excel.



You can review the list of steps that you took to restructure and correct your data in the **Query Settings** pane. When you have completed all steps that you want to take, you can select **Close & Apply** to close Power Query Editor and apply your changes to your data model. However, you can take further action to clean and transform your data.

Replace null values

Occasionally, you might find that your data sources contain null values. For example, a freight amount on a sales order might have a null value if it's synonymous with zero. If the value stays null, the averages will not calculate correctly. One solution would be to change the nulls to zero, which will produce the more accurate freight average. In this instance, using the same steps that you followed previously will help you replace the null values with zero.

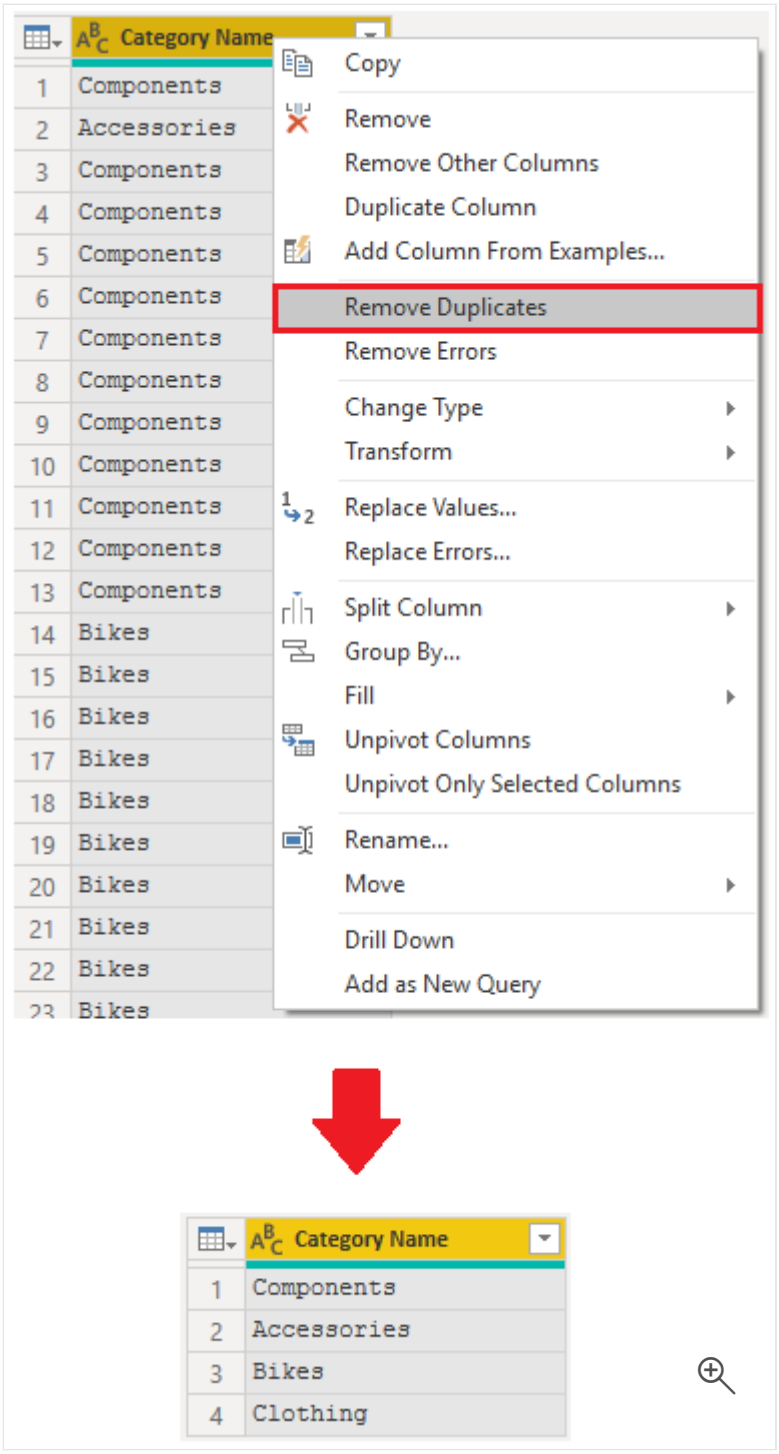


Remove duplicates

You can also remove duplicates from columns to only keep unique names in a selected column by using the **Remove Duplicates** feature in Power Query.

In this example, notice that the **Category Name** column contains duplicates for each category. As a result, you want to create a table with unique categories and use it in your data model. You can achieve this action by selecting a column, right-clicking on the header of the column, and then selecting the **Remove Duplicates** option.

You might consider copying the table before removing the duplicates. The **Copy** option is at the top of the context menu, as shown in the following screenshot. Copying the table before removing duplicates will give you a comparison of the tables and will let you use both tables, if needed.



Best practices for naming tables, columns, and values

Naming conventions for tables, columns, and values have no fixed rules; however, we recommend that you use the language and abbreviations that are commonly used within your organization and that everyone agrees on and considers them as common terminology.

A best practice is to give your tables, columns, and measures descriptive business terms and replace underscores ("_") with spaces. Be consistent with abbreviations, prefixes, and words like "number" and "ID." Excessively short abbreviations can cause confusion if they are not commonly used within the organization.

Also, by removing prefixes or suffixes that you might use in table names and instead naming them in a simple format, you will help avoid confusion.

When replacing values, try to imagine how those values will appear on the report. Values that are too long might be difficult to read and fit on a visual. Values that are too short might be difficult to interpret. Avoiding acronyms in values is also a good idea, provided that the text will fit on the visual.

Next unit: Evaluate and change column data types

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Evaluate and change column data types

10 minutes

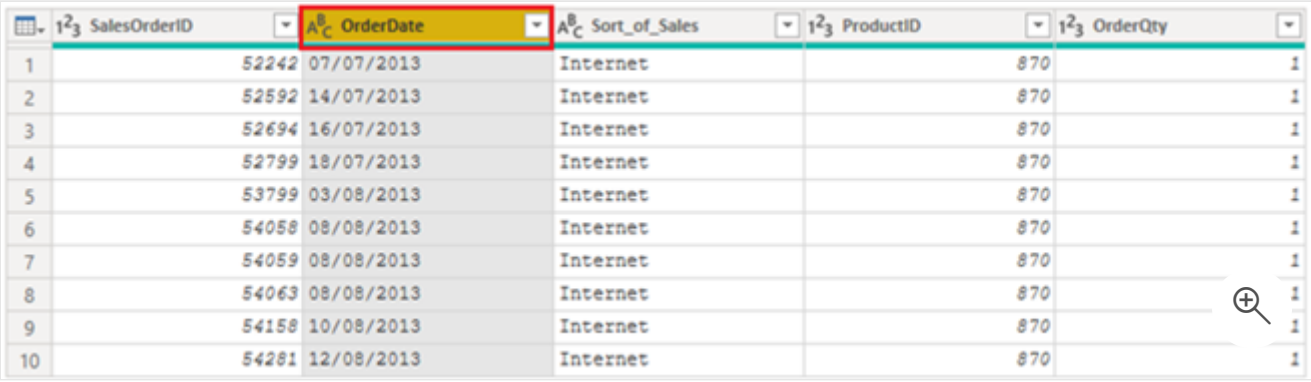
When you import a table from any data source, Power BI Desktop automatically starts scanning the first 1,000 rows (default setting) and tries to detect the type of data in the columns. Some situations might occur where Power BI Desktop doesn't detect the correct data type. Where incorrect data types occur, you'll experience performance issues.

You have a higher chance of getting data type errors when you're dealing with flat files, such as comma-separated values (.CSV) files and Excel workbooks (.XLSX), because data was entered manually into the worksheets and mistakes were made. Conversely, in databases, the data types are predefined when tables or views are created.

A best practice is to evaluate the column data types in Power Query Editor before you load the data into a Power BI data model. If you determine that a data type is incorrect, you can change it. You might also want to apply a format to the values in a column and change the summarization default for a column.

To continue with the scenario where you're cleaning and transforming sales data in preparation for reporting, you now need to evaluate the columns to ensure that they have the correct data type. You need to correct any errors that you identify.

You evaluate the **OrderDate** column. As expected, it contains numeric data, but Power BI Desktop has incorrectly set the column data type to Text. To report on this column, you need to change the data type from Text to Date.



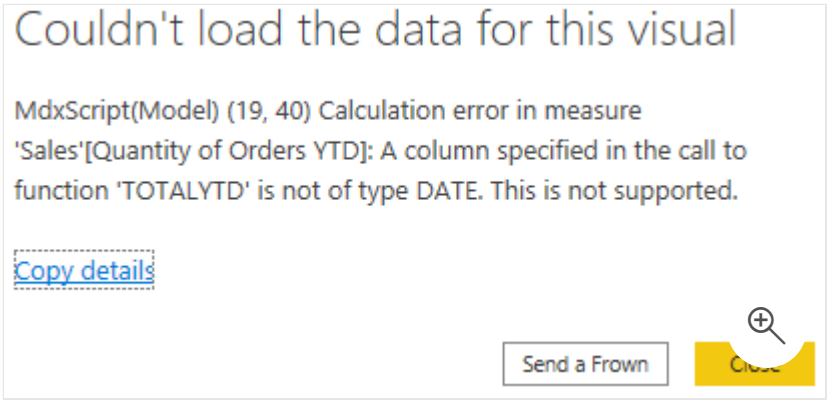
	SalesOrderID	OrderDate	Sort_of_Sales	ProductID	OrderQty
1	52242	07/07/2013	Internet	870	1
2	52592	14/07/2013	Internet	870	1
3	52694	16/07/2013	Internet	870	1
4	52799	18/07/2013	Internet	870	1
5	53799	03/08/2013	Internet	870	1
6	54058	08/08/2013	Internet	870	1
7	54059	08/08/2013	Internet	870	1
8	54063	08/08/2013	Internet	870	1
9	54158	10/08/2013	Internet	870	1
10	54281	12/08/2013	Internet	870	1

Implications of incorrect data types

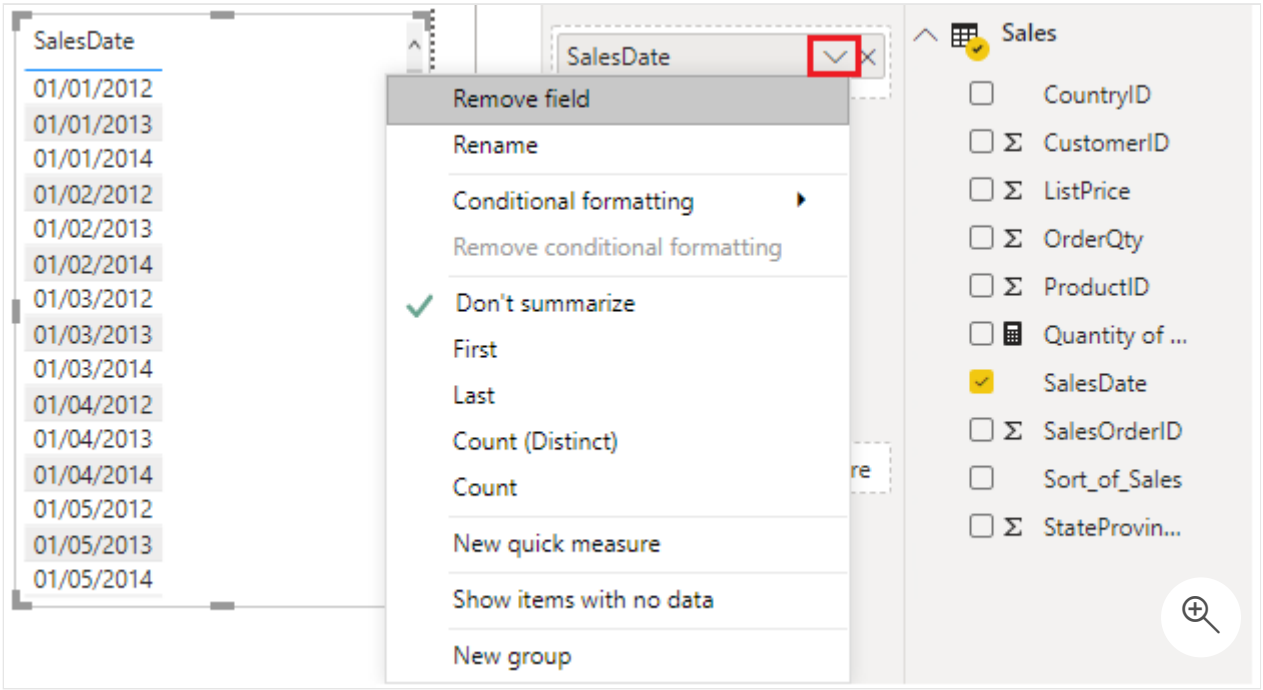
The following information provides insight into problems that can arise when Power BI doesn't detect the correct data type.

Incorrect data types will prevent you from creating certain calculations, deriving hierarchies, or creating proper relationships with other tables. For example, if you try to calculate the Quantity of Orders YTD, you'll get the following error stating that the **OrderDate** column data type isn't Date, which is required in time-based calculations.

```
Quantity of Orders YTD = TOTALYTD(SUM('Sales'[OrderQty]), 'Sales'[OrderDate])
```



Another issue with having an incorrect data type applied on a date field is the inability to create a date hierarchy, which would allow you to analyze your data on a yearly, monthly, or weekly basis. The following screenshot shows that the SalesDate field isn't recognized as type Date and will only be presented as a list of dates in the Table visual. However, It's a best practice to use a date table and turn off the auto date/time to get rid of the auto generated hierarchy. For more information about this process, see Auto generated data type documentation.

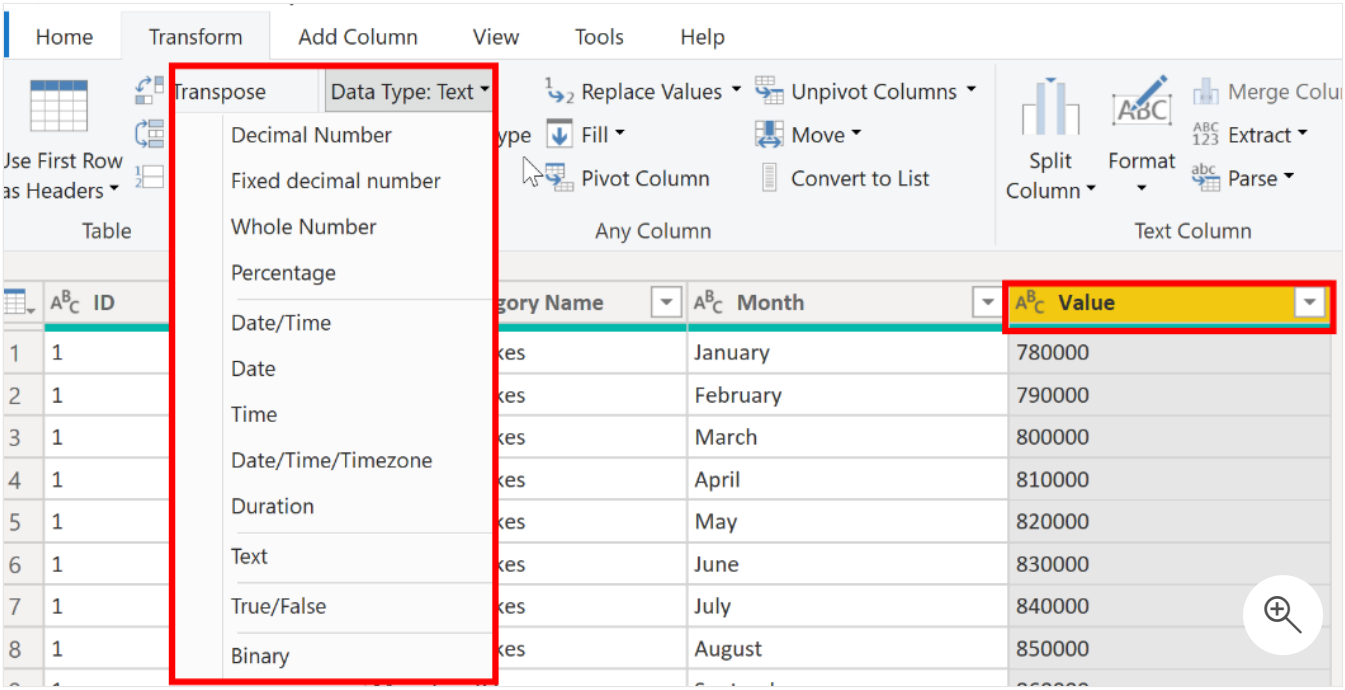


Change the column data type

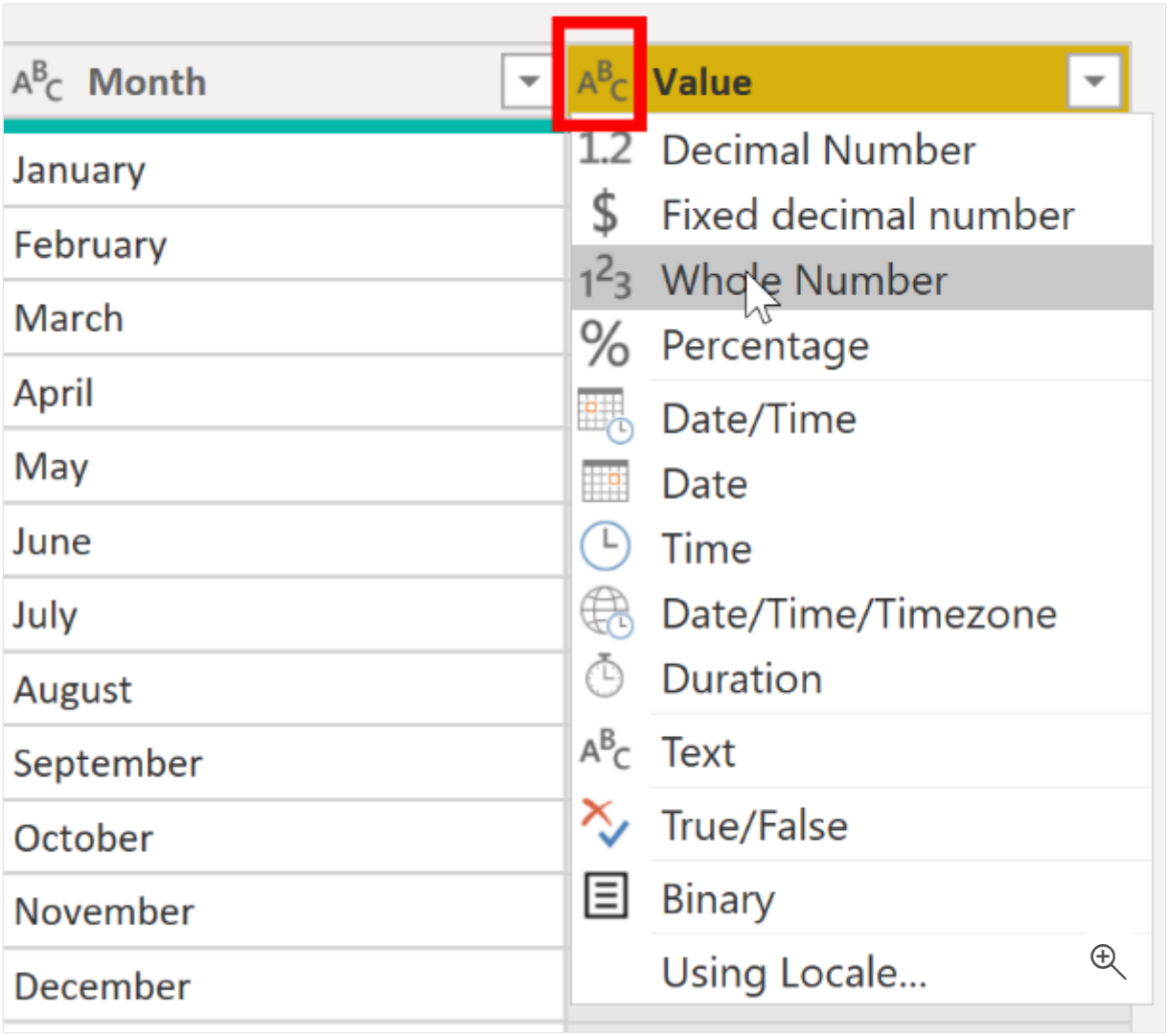
You can change the data type of a column in two places: in Power Query Editor and in the Power BI Desktop Report view by using the column tools. It is best to change the data type in the Power Query Editor before you load the data.

Change the column data type in Power Query Editor

In Power Query Editor, you can change the column data type in two ways. One way is to select the column that has the issue, select **Data Type** in the **Transform** tab, and then select the correct data type from the list.



Another method is to select the data type icon next to the column header and then select the correct data type from the list.



As with any other changes that you make in Power Query Editor, the change that you make to the column data type is saved as a programmed step. This step is called **Changed Type** and it will be iterated every time the data is refreshed.

After you have completed all steps to clean and transform your data, select **Close & Apply** to close Power Query Editor and apply your changes to your data model. At this stage, your data should be in great shape for analysis and reporting.

For more information, see Data types in Power BI Desktop.

Next unit: Combine multiple tables into a single table

Continue >

100 XP

Combine multiple tables into a single table

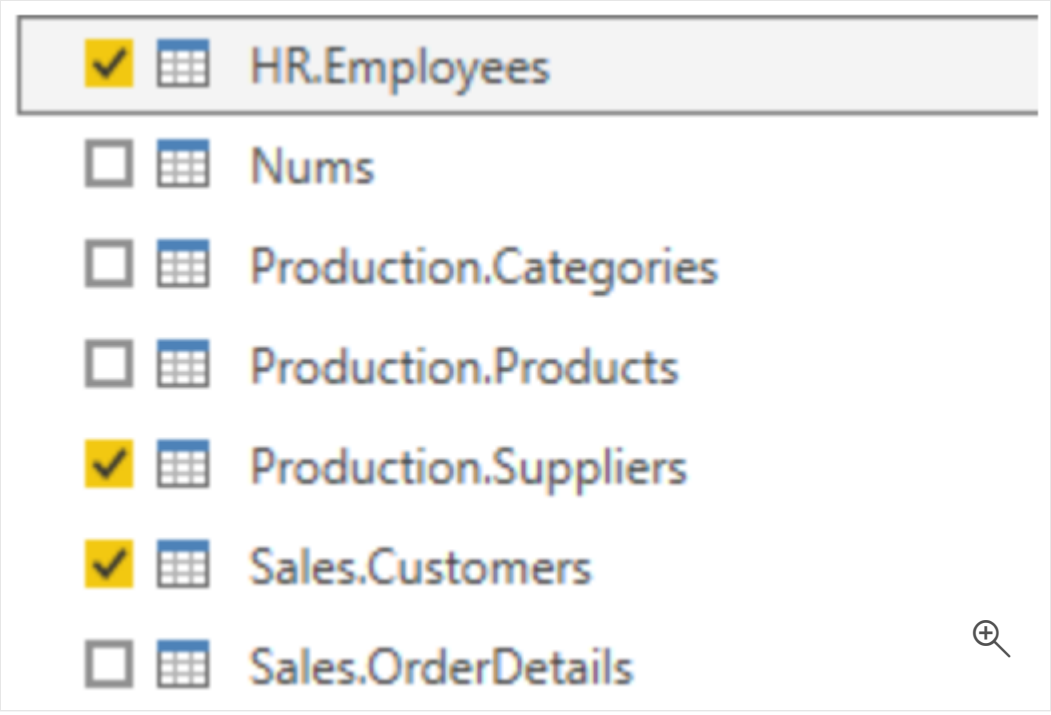
8 minutes

The ability to combine queries is powerful because it allows you to append or merge different tables or queries together. You can combine tables into a single table in the following circumstances:

- Too many tables exist, making it difficult to navigate an overly complicated data model.
- Several tables have a similar role.
- A table has only a column or two that can fit into a different table.
- You want to use several columns from different tables in a custom column.

You can combine the tables in two different ways: merging and appending.

Assume that you're developing Power BI reports for the Sales and HR teams. They have asked you to create a contact information report that contains the contact information and location of every employee, supplier, and customer. The data is in the HR.Employees, Production.Suppliers, and the Sales. Customers tables, as shown in the following image.



However, this data comes from multiple tables, so the dilemma is determining how you can merge the data in these multiple tables and create one source-of-truth table to create a report

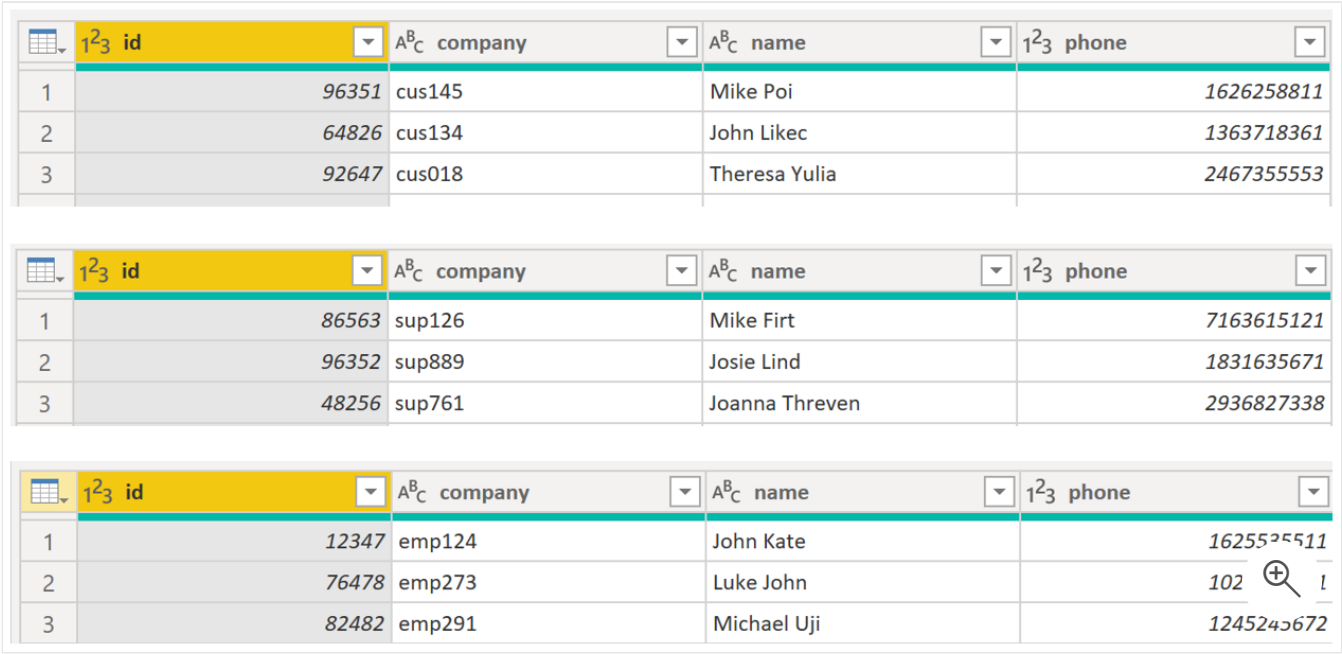
from. The inherent functionality of Power BI allows you to combine and merge queries into a single table.

Append queries

When you append queries, you'll be adding rows of data to another table or query. For example, you could have two tables, one with 300 rows and another with 100 rows, and when you append queries, you'll end up with 400 rows. When you merge queries, you'll be adding columns from one table (or query) into another. To merge two tables, you must have a column that is the key between the two tables.

For the previously mentioned scenario, you'll append the HR.Employees table with the Production. Suppliers and Sales. Customers tables so that you have one master list of contact information. Because you want to create one table that has all contact information for employees, suppliers, and customers, when you combine the queries, the pertinent columns that you require in your combined table must be named the same in your original data tables to see one consolidated view.

Before you begin combining queries, you can remove extraneous columns that you don't need for this task from your tables. To complete this task, format each table to have only four columns with your pertinent information, and rename them so they all have the same column headers: ID, company, name, and phone. The following images are snippets of the reformatted Sales.Customers, Production. Suppliers, and HR.Employees tables.



	id	company	name	phone
1	96351	cus145	Mike Poi	1626258811
2	64826	cus134	John Likec	1363718361
3	92647	cus018	Theresa Yulia	2467355553

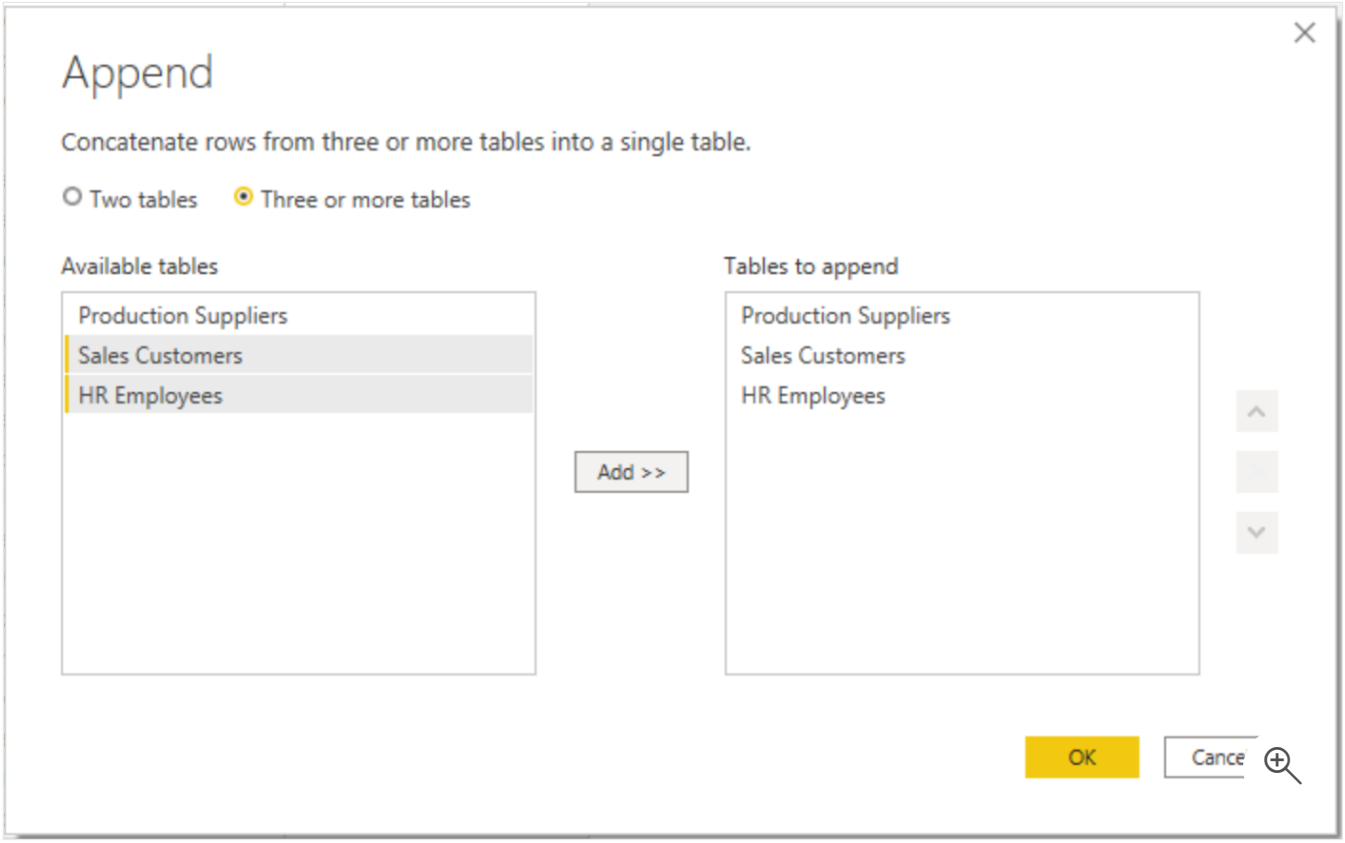
	id	company	name	phone
1	86563	sup126	Mike Firt	7163615121
2	96352	sup889	Josie Lind	1831635671
3	48256	sup761	Joanna Threven	2936827338

	id	company	name	phone
1	12347	emp124	John Kate	1625535511
2	76478	emp273	Luke John	102
3	82482	emp291	Michael Uji	1245245672

After you have finished reformatting, you can combine the queries. On the **Home** tab on the Power Query Editor ribbon, select the drop-down list for **Append Queries**. You can select **Append Queries as New**, which means that the output of appending will result in a new query

or table, or you can select **Append Queries**, which will add the rows from an existing table into another.

Your next task is to create a new master table, so you need to select **Append Queries as New**. This selection will bring you to a window where you can add the tables that you want to append from **Available Tables** to **Tables to Append**, as shown in the following image.



After you have added the tables that you want to append, select **OK**. You'll be routed to a new query that contains all rows from all three of your tables, as shown in the following image.

	id	company	name	phone
1	12347	emp124	John Kate	1625535511
2	76478	emp273	Luke John	1028777351
3	82482	emp291	Michael Uji	1245245672
4	97436	emp173	Kate Fitch	2352467634
5	12462	emp270	Eve Jun	3578999554
6	35237	emp715	Don Potre	3579006677
7	23467	emp183	Marc Wetb	2245789954
8	13892	emp163	Sara Scotts	2388367234
9	56356	emp172	Mitch Potter	1234683673
10	23478	emp812	Liliy Kithc	4567800522
11	45783	emp818	Ren Swrte	2357997515
12	86563	sup126	Mike Firt	7163615121
13	96352	sup889	Josie Lind	1831635671
14	48256	sup761	Joanna Threven	2936827338
15	28461	sup163	Michael Bob	1937293165
16	83613	sup162	Mimi Jukth	2916384462
17	96351	cus145	Mike Poi	1626258811
18	64826	cus134	John Likec	1363718361
19	92647	cus018	Theresa Yulia	2467355553
20	91661	cus182	Ren Thibe	3345783234
21	1736	cus104	Ron Mikel	1235[illegible]789
22	1835	cus103	Joy Qui	234 1
23	1745	cus141	Cat Yate	2345678986

You have now succeeded in creating a master table that contains the information for the employees, suppliers, and customers. You can exit Power Query Editor and build any report elements surrounding this master table.

However, if you wanted to merge tables instead of appending the data from one table to another, the process would be different.

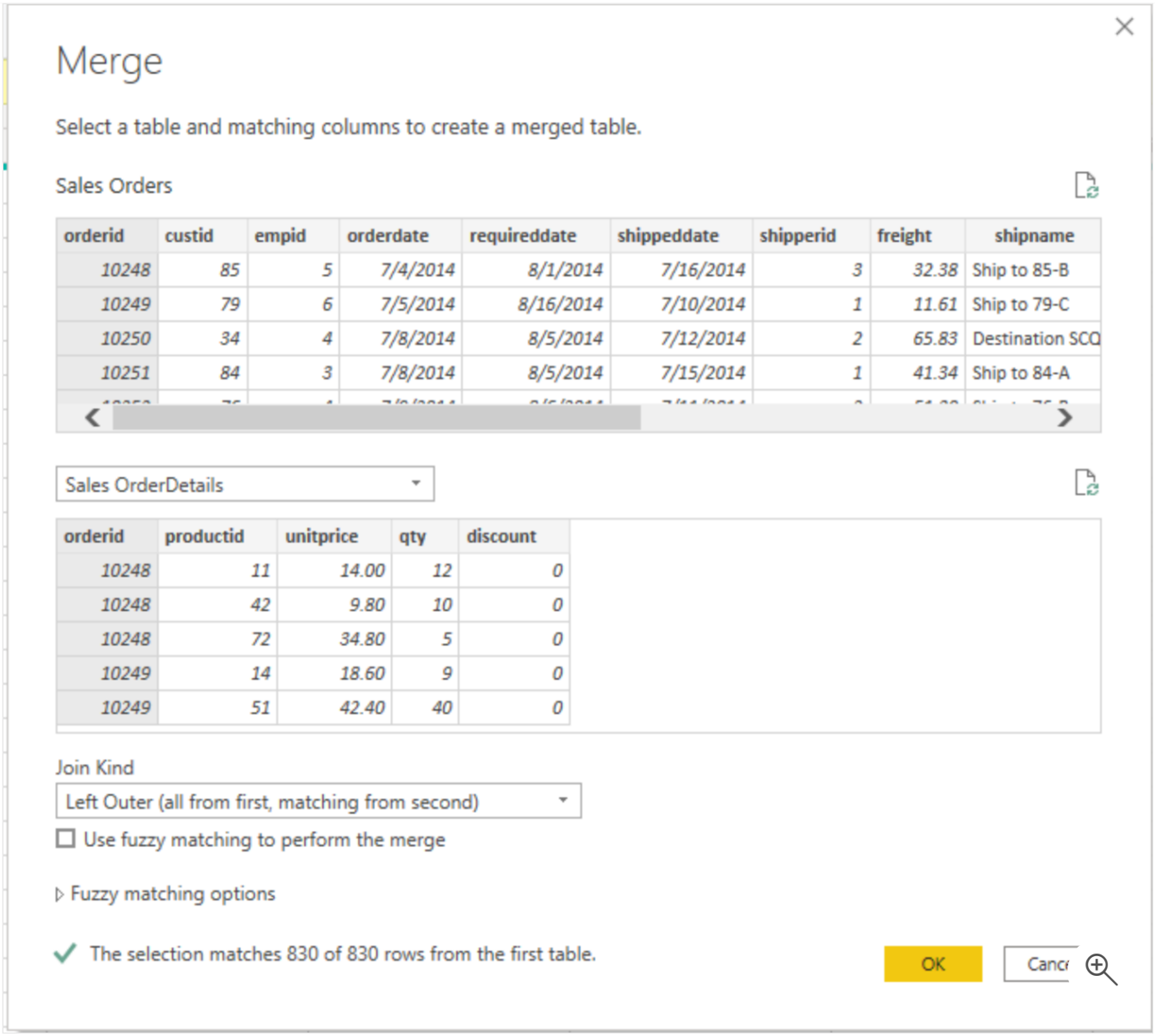
Merge queries

When you merge queries, you're combining the data from multiple tables into one based on a column that is common between the tables. This process is similar to the JOIN clause in SQL. Consider a scenario where the Sales team now wants you to consolidate orders and their corresponding details (which are currently in two tables) into a single table. You can accomplish this task by merging the two tables, Orders and OrderDetails, as shown in the following image. The column that is shared between these two tables is **OrderID**.

	orderid	orderdate	shipperid
1	1	4/23/2018	12
2	2	4/25/2018	24
3	3	6/12/2018	19
4	4	6/13/2018	13
5	5	7/23/2018	11
6	6	7/25/2018	33

	orderid	productid	qty	unitprice
1	1	124	12	14
2	2	134	55	11.2
3	3	641	57	45
4	4	98	5	112.5
5	5	312	23	11.1
6	6	124	78	

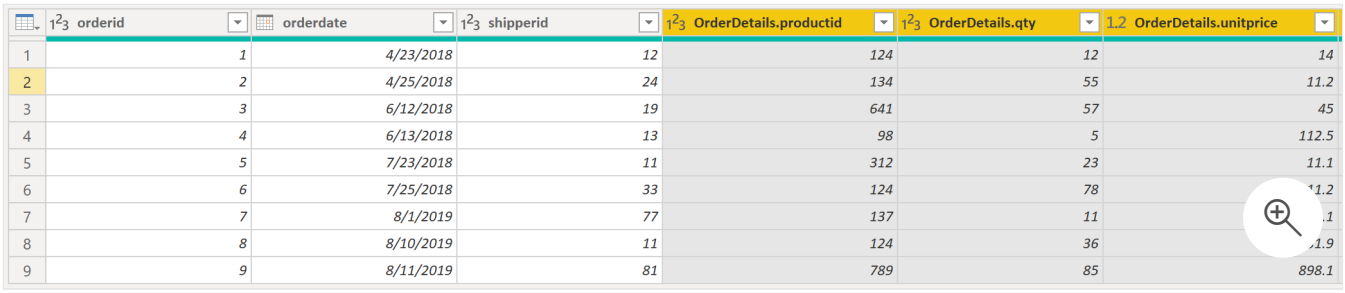
Go to **Home** on the Power Query Editor ribbon and select the **Merge Queries** drop-down menu, where you can select **Merge Queries as New**. This selection will open a new window, where you can choose the tables that you want to merge from the drop-down list, and then select the column that is matching between the tables, which in this case is **ordered**.



You can also choose how to join the two tables together, a process that is also similar to JOIN statements in SQL. These join options include:

- **Left Outer** - Displays all rows from the first table and only the matching rows from the second.
- **Full Outer** - Displays all rows from both tables.
- **Inner** - Displays the matched rows between the two tables.

For this scenario, you'll choose to use a **Left Outer** join. Select **OK**, which will route you to a new window where you can view your merged query.



	orderid	orderdate	shipperid	OrderDetails.productid	OrderDetails.qty	OrderDetails.unitprice
1	1	4/23/2018	12	124	12	14
2	2	4/25/2018	24	134	55	11.2
3	3	6/12/2018	19	641	57	45
4	4	6/13/2018	13	98	5	112.5
5	5	7/23/2018	11	312	23	11.1
6	6	7/25/2018	33	124	78	1.2
7	7	8/1/2019	77	137	11	.1
8	8	8/10/2019	11	124	36	1.9
9	9	8/11/2019	81	789	85	898.1

Now, you can merge two queries or tables in different ways so that you can view your data in the most appropriate way for your business requirements.

For more information on this topic, see the Shape and Combine Data in Power BI documentation.

Next unit: Profile data in Power BI

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Profile data in Power BI

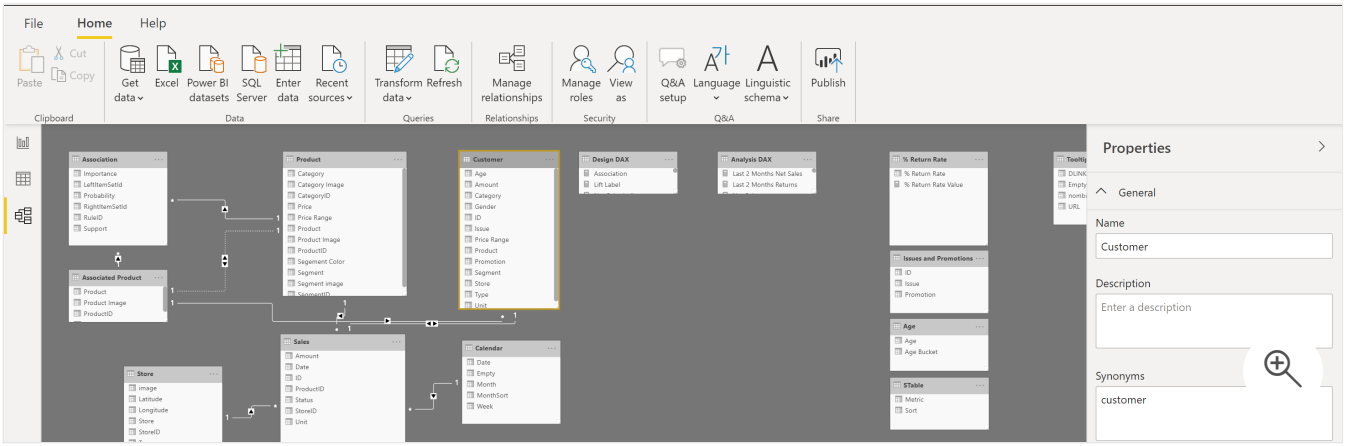
20 minutes

Profiling data is about studying the nuances of the data: determining anomalies, examining and developing the underlying data structures, and querying data statistics such as row counts, value distributions, minimum and maximum values, averages, and so on. This concept is important because it allows you to shape and organize the data so that interacting with the data and identifying the distribution of the data is uncomplicated, therefore helping to make your task of working with the data on the front end to develop report elements near effortless.

Assume that you are developing reports for the Sales team at your organization. You are uncertain how the data is structured and contained within the tables, so you want to profile the data behind the scenes before you begin developing the visuals. Power BI has inherent functionality that makes these tasks user-friendly and straightforward.

Examine data structures

Before you begin examining the data in Power Query Editor, you should first learn about the underlying data structures that data is organized in. You can view the current data model under the **Model** tab on Power BI Desktop.



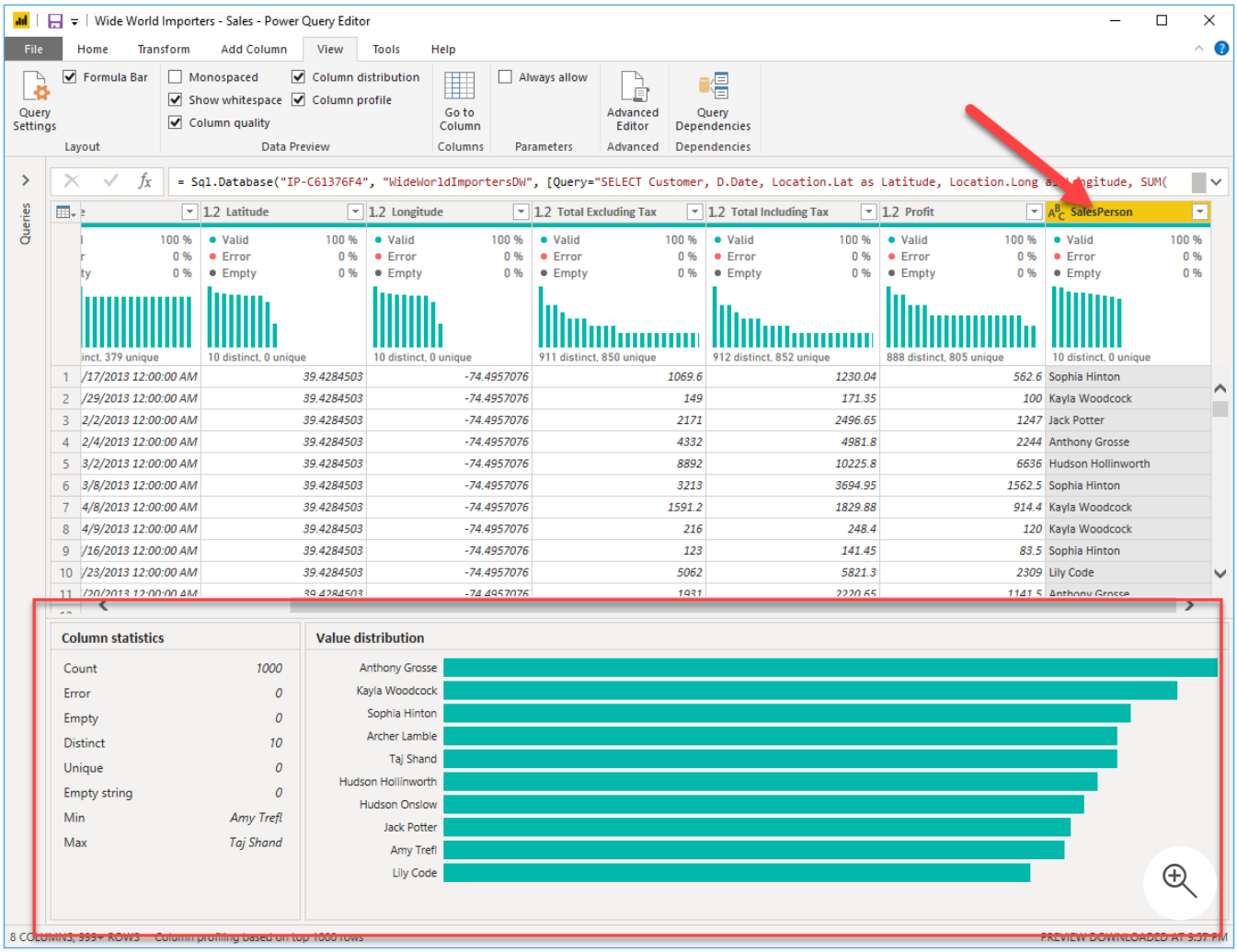
On the **Model** tab, you can edit specific column and table properties by selecting a table or columns, and you can transform the data by using the **Transform Data** button, which takes you to Power Query Editor. Additionally, you can manage, create, edit, and delete relationships between different tables by using **Manage Relationships**, which is located on the ribbon.

Find data anomalies and data statistics

After you have created a connection to a data source and have selected **Transform Data**, you are brought to Power Query Editor, where you can determine if anomalies exist within your data. Data anomalies are outliers within your data. Determining what those anomalies are can help you identify what the normal distribution of your data looks like and whether specific data points exist that you need to investigate further. Power Query Editor determines data anomalies by using the **Column Distribution** feature.

Select **View** on the ribbon, and under **Data Preview**, you can choose from a few options. To understand data anomalies and statistics, select the **Column Distribution**, **Column Quality**, and **Column Profile** options. The following figure shows the statistics that appear.

Column quality and **Column distribution** are shown in the graphs above the columns of data. **Column quality** shows you the percentages of data that is valid, in error, and empty. In an ideal situation, you want 100 percent of the data to be valid.



> **Note**
>
> By default, Power Query examines the first 1000 rows of your data set. To change this, select the profiling status in the status bar and select **Column profiling based on entire data set**.]

Column distribution shows you the distribution of the data within the column and the counts of distinct and unique values, both of which can tell you details about the data counts. Distinct values are all the different values in a column, including duplicates and null values, while unique values do not include duplicates or nulls. Therefore, **distinct** in this table tells you the total count of how many values are present, while **unique** tells you how many of those values only appear once.

Column profile gives you a more in-depth look into the statistics within the columns for the first 1,000 rows of data. This column provides several different values, including the count of rows, which is important when verifying whether the importing of your data was successful. For example, if your original database had 100 rows, you could use this row count to verify that 100 rows were, in fact, imported correctly. Additionally, this row count will show how many rows that Power BI has deemed as being outliers, empty rows and strings, and the min and max, which will tell you the smallest and largest value in a column, respectively. This distinction is particularly important in the case of numeric data because it will immediately notify you if you have a maximum value that is beyond what your business identifies as a "maximum." This value calls to your attention these values, which means that you can then focus your efforts when delving deeper into the data. In the case where data was in the text column, as seen in the previous image, the minimum value is the first value and the maximum value is the last value when in alphabetical order.

Additionally, the **Value distribution** graph tells you the counts for each distinct value in that specific column. When looking at the graph in the previous image, notice that the value distribution indicates that "Anthony Gross" appears the greatest number of times within the **SalesPerson** column and that "Lily Code" appears the least number of times. This information is particularly important because it identifies outliers. If a value appears far more than other values in a column, the **Value distribution** feature allows you to pinpoint a place to begin your investigation into why this is so.

On a numeric column, **Column Statistics** will also include how many zeroes and null values exist, along with the average value in the column, the standard deviation of the values in the column, and how many even and odd values are in the column. These statistics give you an idea of the distribution of data within the column, and are important because they summarize the data in the column and serve as a starting point to determine what the outliers are.

For example, while looking through invoice data, you notice that the **Value distribution** graph shows that a few salespeople in the **SalesPerson** column appear the same number of times within the data. Additionally, you notice the same situation has occurred in the **Profit** column and in a few other tables as well. During your investigation, you discover that the data you were using was bad data and needed to be refreshed, so you immediately complete the refresh. Without viewing this graph, you might not have seen this error so quickly and, for this reason, value distribution is essential.

After you have completed your edits in Power Query Editor and are ready to begin building visuals, return to **Home** on the Power Query Editor ribbon. Select **Close & Apply**, which will return you to Power BI Desktop and any column edits/transformations will also be applied.

You have now determined the elements that make up profiling data in Power BI, which include loading data in Power BI, interrogating column properties to gain clarity about and make further edits to the type and format of data in columns, find data anomalies, and view data statistics in Power Query Editor. With this knowledge, you can include in your toolkit the ability to study your data in an efficient and effective manner.

Next unit: Use Advanced Editor to modify M code

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

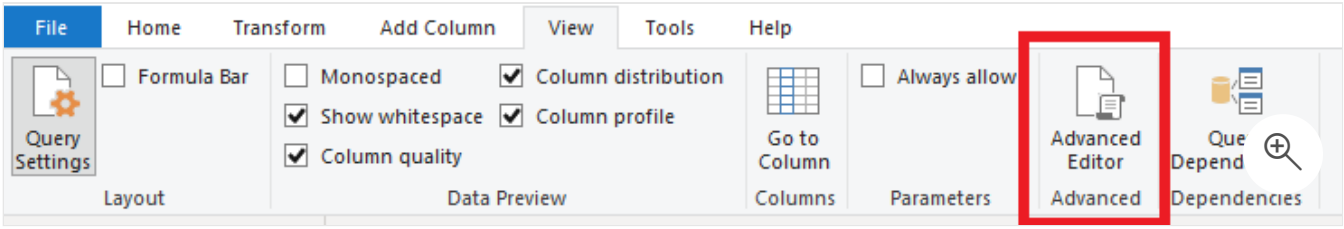
100 XP

Use Advanced Editor to modify M code

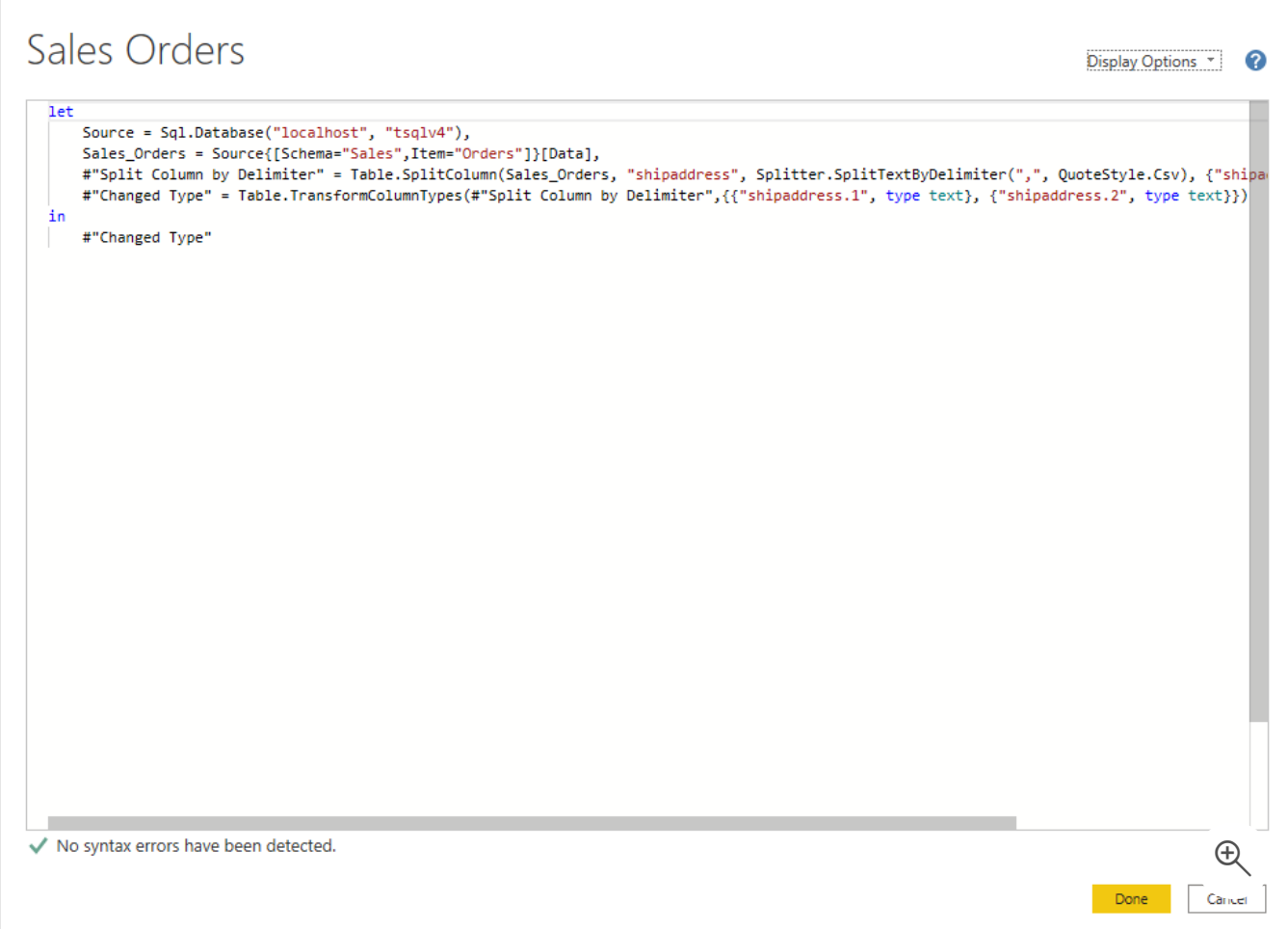
4 minutes

Each time you shape data in Power Query, you create a step in the Power Query process. Those steps can be reordered, deleted, and modified where it makes sense. Each cleaning step that you made was likely created by using the graphical interface, but Power Query uses the M language behind the scenes. The combined steps are available to read by using the Power Query Advanced Editor. The M language is always available to be read and modified directly. It is not required that you use M code to take advantage of Power Query. You will rarely need to write M code, but it can still prove useful. Because each step in Power Query is written in M code, even if the UI created it for you, you can use those steps to learn M code and customize it to suit your needs.

After creating steps to clean data, select the **View** ribbon of Power Query and then select **Advanced Editor**.



The following screen should appear.



Each Power Query step will roughly align with one or two lines of M code. You don't have to be an expert in M code to be able to read it. You can even experiment with changing it. For instance, if you need to change the name of a database, you could do it right in the code and then select **Done**.

You might notice that M code is written top-down. Later steps in the process can refer to previous steps by the variable name to the left of the equal sign. Be careful about reordering these steps because it could ruin the statement dependencies. Write to a query formula step by using the **in** statement. Generally, the last query step is used as the **in final data set** result.

Next unit: Exercise - Load data in Power BI Desktop

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

200 XP

Check your knowledge

5 minutes

Answer the following questions to see what you've learned.

1. What is a risk of having null values in a numeric column? *

- That function SUM of data are incorrect.
- That function MAX of data will be incorrect.

  ✗ This is incorrect. MAX expressions will pick the maximum non-null value.
- That function AVERAGE of data will be incorrect.

  ✓ Correct. AVERAGE takes the total and divides by the number of non-null values. If NULL is synonymous with zero in the data, the average will be different from the accurate average.

2. If you have two queries that have different data but the same column headers, and you want to combine both tables into one query with all the combined rows, which operation should you perform? *

- Append

  ✓ Correct. Append will take two tables and combine it into one query. The combined query will have more rows while keeping the same number of columns.
- Merge
- Combine Column

  ✗ Incorrect. This is a column operation, not a table operation.

3. Which of the following selections aren't best practices for naming conventions in Power BI? *

- ◯ Rename columns to have spaces in them.

  ✗ **Incorrect. Spaces make columns and tables easier to read for report users and report authors.**

- ◯ Abbreviate column names.

  ✓ **Correct. Abbreviations lead to confusion because they're often overused or not universally agreed on.**

- ◯ Replace values that have integers with human readable results.

Next unit: Summary

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

Practice Assessment for Exam PL-300: Microsoft Power BI Data Analyst

Question 21 of 50

You import an Excel file into Power BI Desktop and begin to analyze the data in Power Query Editor.

You need to identify outliers in a text column within the data source.

Which information should you use from Power Query Editor?

- ◯ the min and max values in Column profile
- ◯ the top and bottom entries in Value distribution

✓This answer is correct.

- ◯ the value of the Distinct entry in Column statistics

This answer is incorrect.

- ◯ the value of the Unique entry in Column statistics

The top and bottom entries in Value distribution identify outliers, which appear, respectively, the greatest and the smallest number of times in that column. The value of the Distinct entry in Column statistics indicates the total count of different values. The value of the Unique entry in Column statistics indicates the total count of different values that appear only once. The min and max values in the Column profile of a text column designate the entries which appear, respectively, first and last in alphabetical order.

Profile data in Power BI - Training | Microsoft Learn

Next >

Check Your Answer

Question 22 of 50

You use Power BI Desktop to transform a query that retrieves data from an Excel spreadsheet.

You need to view and modify multiple steps in all the M code.

Which interface should you use?

- ◯ Advanced Editor in the Power Query Editor

✓This answer is correct.

- ◯ Data view
- ◯ Formula Bar in the Power Query Editor
- ◯ Model view

The Advanced Editor in the Power Query Editor provides the ability to work with all the M code used for shaping data in a query at once. Data view provides access to data within a dataset. Model view provides access to the data model of a dataset. The Formula Bar in the Power Query Editor provides the ability to edit only one step at a time.

Use Advanced Editor to modify M code - Training | Microsoft Learn

Next >

Check Your Answer

Practice Assessment for Exam PL-300: Microsoft Power BI Data Analyst

Question 23 of 50

You have an Excel spreadsheet that contains three columns labeled Year, 2021, and 2012. The entries in rows for the first column consists of names of the individual months in the year while the other two columns contain the sales amount for each month for the corresponding year.

You import data from the Excel spreadsheet into Power BI Desktop.

You need to transform the data so it will consist of three columns, with the first one containing month, the second containing year, and the third containing the sales amount for that month and year.

Which transformation should you use first?

- ○ Pivot
- ○ Remove Columns
- ○ Transpose Table
- ◉ Unpivot

✓ **This answer is correct.**

Selecting Unpivot will allow you to shape the current table into the one with the year, month, and sales amount columns, which will need to be renamed afterwards. Pivot would be applicable in the opposite scenario, in which flat data needs to be reorganized into one containing aggregate values for each unique value in each column. Transposing would switch rows and columns. Removing columns would result in a table with insufficient data to perform unpivot.

Shape the initial data - Training | Microsoft Learn

Unpivot columns - Power Query | Microsoft Learn

Practice Assessment for Exam PL-300: Microsoft Power BI Data Analyst

Question 24 of 50

You have a query named FactSales that retrieves data from a SQL Server table.

You need to update the query to prevent new columns that may be added to the table in the future from getting imported during dataset refreshes.

You select the existing columns in the query.

Which two applied steps can you use in Power Query Editor to achieve the goal? Each correct answer presents a complete solution.

- [x] Choose Columns

✓This answer is correct.

- [ ] Remove Columns
- [x] Remove Other Columns

✓This answer is correct.

- [ ] Transpose

Using the Choose Columns and Remove Other Columns options will allow you to explicitly select the columns that you want to keep. This enforces the requirement that new columns will not be automatically added in the future. Using Remove Columns will delete the existing columns in the table, and new columns that get added in the future will still be imported automatically. Transpose treats rows as columns, and columns as rows. It will not limit the number of columns being imported.

Shape the initial data - Training | Microsoft Learn

Choose or remove columns - Power Query | Microsoft Learn

Practice Assessment for Exam PL-300: Microsoft Power BI Data Analyst

Question 25 of 50

From the Power Query Editor, you import data from a .csv file. The data includes a column named ZIP that contains postal codes from the United States.

You notice that Power Query Editor automatically applies the Whole Number data type to the ZIP column.

You need to ensure that the ZIP column uses the Text data type and that all values remain 5 characters long.

What should you do?

- ○ From the data view in Power BI Desktop, change the column data type from number to text.
- ◯ From Power Query Editor, add a new applied step at the end of the query to convert the ZIP column from number back to text.

This answer is incorrect.

- ○ From Power Query Editor, delete the changed type step.
- ◯ From Power Query Editor, update the current changed type step and replace convert from number to text for the ZIP column.

✓**This answer is correct.**

To correctly update the data to text you need to replace the number type conversion with a text conversion, and to keep all other data type column transformations. This needs to be done in the Power Query Editor. Adding a new applied step at the end of the query would result in loosing zip codes that start with 0. Changing the data type in Data View is equivalent to adding an applied step at the end of the query and would not preserve leading zeros. Deleting the changed type step would not set the data type to Text.

Evaluate and change column data types - Training | Microsoft Learn

Question 20 of 50

You have a query that retrieves data from a table that contains more than 8,000 rows of data.

In Power Query Editor, you notice that the column statistics for each column shows a count of exactly 1,000.

You need to ensure that the column statistics for each column shows the statistics based on all rows that are returned by the query.

What should you do?

- ◯ Add a `Table.Buffer` applied step to the query.
- ◯ Change the query load type from Import to **DirectQuery**.
- ◯ Create a Top N row count parameter for the query.
- ◯ From the query window, select **Column profiling on the entire dataset**.

✓**This answer is correct.**

Selecting column profiling for the entire dataset will change the column profiler to analyze the entire query dataset. Adding a `Table.Buffer` applied step will only cache the entire query in memory during refresh. The query load type in the model will not impact how many rows are used by the column profiler. Using a Top N row count applied step or applying a parameter will not change how the column statistics feature calculates.

Profile data in Power BI - Training | Microsoft Learn

Using the data profiling tools - Power Query | Microsoft Learn

Practice Assessment for Exam PL-300: Microsoft Power BI Data Analyst

Question 21 of 50

You are analyzing query data by using Power Query Editor.

You need to ensure that the Column statistics are based on an analysis of the entire dataset.

What should you do?

- ◯ Change the query connection type from Import to DirectQuery.
- ◯ From the status bar, change profiling status to entire dataset.

  ✓This answer is correct.
- ◯ In Power Query Editor, enable column profiling from the View ribbon.

  This answer is incorrect.
- ◯ Load the data into the data model.

By default, column profiling is only based on the first 1000 rows of the preview. Changing the column profile status to entire dataset will query the entire dataset for the column profiling information.

Profile data in Power BI - Training | Microsoft Learn

Next > Check Your Answer

Practice Assessment for Exam PL-300: Microsoft Power BI Data Analyst

Question 23 of 50

You connect Power Query Editor to a database table.

You need to remove the Row ID column. Your solution must ensure that new columns do NOT display in the table model during a scheduled refresh in the future.

What transformation should you use?

- ◯ Select **Row ID**, then use the **Remove Other Columns** command.
- ◯ Use the **Remove Column** command on the **Row ID** column.

  This answer is incorrect.
- ◯ Use the **Select Columns** command and chose the columns to keep.

  ✓**This answer is correct.**
- ◯ Use the **Transpose** command, then filter the rows to remove **Row ID**.

Only the **Select Columns** command will let you choose columns to keep, delete the columns you do not want, and prevent new columns from showing up in the table in the future.

Shape the initial data - Training | Microsoft Learn

Next > Check Your Answer

Practice Assessment for Exam PL-300: Microsoft Power BI Data Analyst

Question 24 of 50

You have a Power BI data source that contains the following tables:

- ProductCategory: Lists the product category ID and category name
- ProductSubcategory: Lists the product subcategory ID, subcategory name, and the related category ID
- ProductList: Lists the product ID, name, and subcategory ID

You need to optimize the data to create a dimension for use in a star schema data model.

How should the tables be transformed before loading into the model?

- ◯ Import all three tables into the data model and connect them using relationships.

  This answer is incorrect.
- ◯ Import only the ProductName table into the model.
- ◯ Merge the queries to create a single loaded table for Product.

  ✓**This answer is correct.**
- ◯ Use the append command to create a single loaded table for product.

A star schema should have a single table for each dimension or product, so using the combine command is what is required to create a single product table and aim towards a star schema design.
Combine multiple tables into a single table - Training | Microsoft Learn

Next > Check Your Answer

Practice Assessment for Exam PL-300: Microsoft Power BI Data Analyst

Question 26 of 50

You have an Excel spreadsheet that contains two columns as follows:

- Category: Contain names of categories
- Subcategory: Contain names of subcategories for each category

You import the Excel spreadsheet into Power BI Desktop.

You need to transform the data to meet the following requirements:

- Have multiple columns, where one column represents one category
- Have a single row that includes the total count of subcategories for each category

Which transformation should you use?

- ◯ Pivot Columns

  ✓**This answer is correct.**
- ◯ Rename Columns
- ◯ Transpose Table
- ◯ Unpivot Columns

  This answer is incorrect.

The pivot column operation converts data into a table by aggregating values in a column. In this case, you can pivot the columns to calculate the count of product subcategories in each product category. The unpivot column operation performs the opposite task, converting columns into rows. Renaming columns would result in a table with different column names but the same data content. Transposing would simply switch rows and columns, without the required aggregation.

Shape the initial data - Training | Microsoft Learn

Question 27 of 50

You have a fact table that contains sales data.

The fact table includes a SalesDate column formatted as a Date data type. Auto date/time setting is disabled in both global and current file options.

You load the fact table into Power BI Desktop.

You need to ensure that you are able to analyze data on a yearly, quarterly, monthly, weekly, and daily basis. Your solution must minimize the model size and administrative effort.

What should you do?

- ◯ Add a separate date dimension table.

  ✓This answer is correct.
- ◯ Add a year, month, and week columns to the fact table.
- ◯ Enable the Auto date/time current file option.
- ◯ Enable the Auto date/time global option.

Adding a separate date dimension table that includes year, month, and week information is the optimal approach which provides the required functionality, while minimizing the model size and administrative effort. Adding a year, month, and week columns to the fact table would increase the amount of administrative effort. Enabling the Auto date/time global or current file option would increase the model size. In addition, it would not provide the ability to describe weekly time periods (only year, quarter, month, and day).

Evaluate and change column data types - Training | Microsoft Learn

Auto date/time guidance in Power BI Desktop - Power BI | Microsoft Learn

Next >

Check Your Answer

Practice Assessment for Exam PL-300: Microsoft Power BI Data Analyst

Question 28 of 50

You have an Azure SQL database that contains two tables named SalesOrders and SalesOrderDetails.

You load the SalesOrders and SalesOrderDetails tables into Power BI Desktop. The tables have a relationship based upon the SalesOrderID column.

You need to combine the two tables into one table. The solution must meet the following requirements:

- There is a row for every SalesOrderDetails row in the resulting table, even if there is no corresponding SalesOrders row.
- For every SalesOrderDetails row, the row in the resulting table includes the corresponding SalesOrders row data if the SalesOrder row exists.
- Any SalesOrder row data that does not have corresponding SalesOrderDetails rows is not included in the resulting table.

What should you do to combine the two tables?

- ◯ Append the tables.
- ◯ Merge the tables using the full outer join kind based on the sales order id.
- ◯ Merge the tables using the inner join kind based on the sales order id.

This answer is incorrect.

- ◯ Merge the tables using the left outer join kind based on the sales order id.

✓This answer is correct.

To construct the resulting table, it is necessary to merge tables using the left outer join kind with the SalesOrderDetails as the first table. Appending tables in this context would not make sense, since they do not have the same columns. Merging using the inner join kind would not produce a table that includes a sales order details row where there is no corresponding sales order row. Full outer join